Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Maestría en Explotación de Datos y Descubrimiento del conocimiento

Aplicación de modelos de aprendizaje Automático en la eficiencia energética de reciclado de aluminio

Autor: Renso Gil

Resumen: *La eficiencia energética en el reciclado de aluminio es importante para preservar el medio ambiente. La producción primaria de aluminio requiere mucha energía y tiene impactos ambientales negativos. Por ello, el reciclado del aluminio se presenta como una solución prometedora, ya que reduce la necesidad de producción primaria y las emisiones de carbono. El estudio se centra en analizar los hornos de fundición de la planta de Abasto, propiedad de la empresa Aluar, líder en la producción de aluminio. Aluar se dedica a la fundición de aluminio, reciclando el material descartado en distintos procesos productivos y desperdicios de clientes. Se plante como objetivo predecir el consumo de metros cúbicos de gas por tonelada utilizando los métodos de aprendizaje automático y identificar las variables que tienen un mayor impacto en el consumo de gas del proceso. Para cumplir con el objetivo se desarrollaron las siguientes técnicas. Se utilizó análisis exploratorio de datos relevantes en el consumo de gas en hornos de fundir. Se emplea visualización para identificar patrones y relaciones entre el consumo de gas y otras variables relevantes. Se utilizaron los algoritmos Random Forest y XGBoost para la predicción. Para obtener los mejores hiperparámetros se utilizó la optimización Bayesiana. Para comparar los modelos se utilizaron La curva ROC, el AUC-ROC y la Matriz de Confusión. Para agregarle explicatividad a los modelos se utilizó IBreakDown. Se destaca que XGBoost muestra mejor rendimiento según el área bajo la curva ROC, pero Random Forest es más preciso para procesos con alto consumo de gas. Al utilizar IBreakdown para explicar el impacto de variables, se encontró que los kilogramos de carga, tiempo neto, tiempo de quemadores, ubicación del horno, temperatura del baño líquido y mantenimiento influyen en el consumo de gas . Se recomienda trabajar con hornos llenos, analizar procesos con tiempo extendido y respetar el mantenimiento. Se mencionan limitaciones debido a la cantidad de datos y variables, proponiendo adquirir más datos, agregar variables importantes y explorar diferentes modelos para mejorar futuras predicciones.*

Julio 2023

Índice

Índice de figuras

Índice de Tablas

1. Introducción.

La eficiencia energética en el reciclado de aluminio es un tema de gran importancia en la sostenibilidad del medio ambiente. El aluminio es un metal ampliamente utilizado en diversas industrias debido a sus propiedades únicas, como su ligereza, resistencia y capacidad de reciclaje. Sin embargo, su producción primaria a partir de la extracción de bauxita y su posterior refinamiento requieren una cantidad significativa de energía, lo que conlleva emisiones de gases de efecto invernadero y agotamiento de recursos naturales. En este contexto, el enfoque en la eficiencia energética en el reciclado de aluminio se presenta como una solución prometedora [1]. El reciclado de aluminio permite reducir la necesidad de producción primaria y minimizar el impacto ambiental asociado. Además, el proceso de reciclado del aluminio requiere aproximadamente solo el 5 % de la energía necesaria para su producción primaria, lo que resulta en una reducción significativa de las emisiones de carbono [2]. En dicho proceso son aspectos clave la selección y clasificación de chatarra de aluminio, la optimización de procesos de fusión y refinado, así como la recuperación de energía residual y calor.

El estudio tiene como alcance analizar los hornos de fundición de la planta de Abasto, ubicada en La Plata, perteneciente a la empresa Aluar, líder en la producción de alumninio [3]. Aluar se dedica a la elaboración de productos semi-elaborados de aluminio, dentro de su proceso, se dedica a la fundición de alumnio. En la fundición se lleva a cabo el reciclaje del aluminio descartado en distintos procesos productivos, así como los desperdicios provenientes de sus clientes. Además, utilizan lingotes de aluminio puro provenientes de la planta de Puerto Madryn, y se añaden aleantes como hierro, magnesio, manganeso y cobre para dotar al aluminio de propiedades específicas según su composición. El sector encuenta con dos hornos de fundición, cada uno con una capacidad máxima de 12 toneladas. EL primer horno se identifica como estación 1 y el segundo como estación 2. Cada horno cuenta con 2 quemadores que funcionan con gas como combustible. En condiciones normales de operación, cada horno consume aproximadamente 500 metros cúbicos de gas por hora. Cabe destacar que estos dos hornos en conjunto representan el 70 % del consumo total de gas de toda la planta de Aluar.

En el marco de la mejora continua, la gerencia de la empresa establece como objetivo de mejora del consumo de gas el promedio de los 10 mejores meses del último año en términos de metros cúbicos por tonelada procesada. Luego, se compara este resultado con los objetivos de años anteriores y se selecciona el valor más bajo. Según el cálculo previo, el objetivo a cumplir es alcanzar un consumo de 130 metros cúbicos por tonelada procesada. En la figura 1 se observa la evolución de consumo de gas por toneladas procesas de la estación 1 y en la figura 2 de la estación 2.

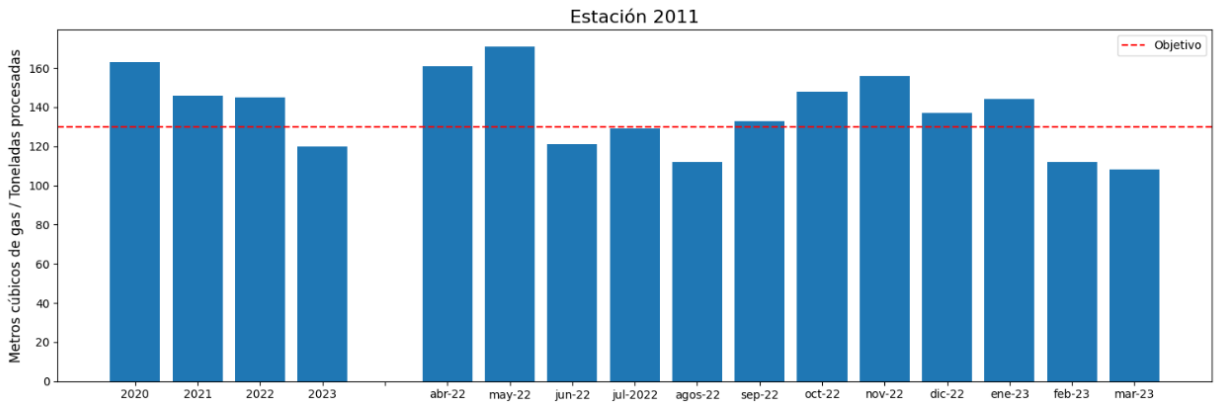

Figura 1: Indicador de consumo de gas por toneladas procesadas de la estación 1

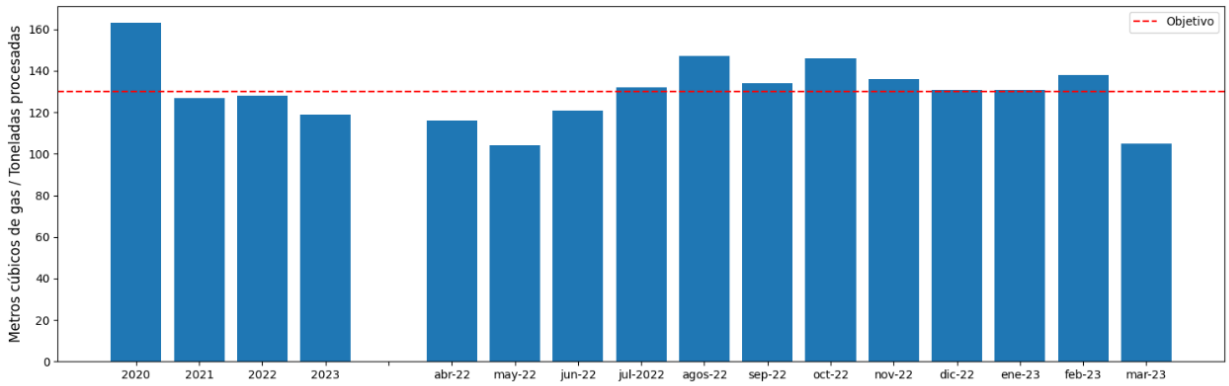

Figura 2: Indicador de consumo de gas por toneladas procesadas de la estación 2

Como objetivos del presente trabajo, se propone lo siguiente:

- Predecir si un proceso va a consumir menos de 130 metros cúbicos por tonelada utilizando los métodos de aprendizaje automático Random forest y XGBoost
- Identificar las variables que tienen un mayor impacto en el consumo de gas del proceso utilizando la herramienta IBreakdown, con el fin de conocer las áreas en las que se pueden realizar modificaciones para reducir la cantidad de gas consumido.

El trabajo se estructura de la siguiente forma. En la primera parte se detalla los conceptos y técnicas de ciencia de datos que se utilizaron. En la segunda parte se explica la metodología de trabajo. Primero se presentarán los datos utilizados y que técnicas se utilizaron en el procesamiento y limpieza. Luego se realiza un análisis exploratorio de datos con las variables más relevantes. A continuación se describe los dos modelos utilizados de aprendizaje automático. Los modelos utilizados son Random Forest y XGBoost, Por ultimo de describe la metodología ibreakdown utilizada para agregarle explicatividad al modelo. En la tercera parte se presentan los resultados obtenidos en ambos modelos utilizando métricas adecuadas y matriz de confusión. Se comparan los valores de ambos modelos. También se presentan los resultados obtenidos al aplicar ibreakdown al mejor modelo. En la cuarta parte se detallan las conclusiones obtenidas, los comentarios finales, También se describen las recomendaciones para futuros estudios.

2. Marco teórico

El proceso de fundición en la industria es una operación crucial que involucra la fusión de metales y materiales para dar forma a diversos productos industriales. En este proceso, los hornos de fundir desempeñan un papel fundamental al proporcionar la fuente de calor necesaria para alcanzar las altas temperaturas requeridas para la fusión de los materiales. La predicción precisa del consumo de gas en estos hornos es de gran importancia, ya que permite optimizar la eficiencia energética y reducir los costos operativos asociados con la producción, además de contribuir a una producción industrial más sostenible y respetuosa con el medio ambiente. El proceso de fundición es un procedimiento crítico en el que los metales sólidos se convierten en líquidos mediante la aplicación de calor. Los hornos de fundir son dispositivos especializados utilizados para llevar a cabo este proceso. Estos hornos funcionan mediante la combustión de gas, generalmente metano, como fuente de energía para calentar los materiales metálicos hasta que alcanzan su punto de fusión y se convierten en líquidos. El consumo de gas en los hornos de fundir depende de varios factores, como el tipo de metal a fundir, el tamaño y capacidad del horno, la temperatura objetivo y la duración del proceso de fundición. La industria enfrenta constantemente la necesidad de mejorar la eficiencia energética y reducir su impacto ambiental. La optimización del consumo de gas en los hornos de fundir es una estrategia esencial para lograr estos objetivos. La reducción del consumo de gas no solo disminuye los costos operativos, sino que también contribuye a la disminución de las emisiones de gases de efecto invernadero, lo que es fundamental para abordar el cambio climático y avanzar hacia una producción más sostenible y responsable [1].

Un ejemplo de aplicación de métodos predictivos para el consumo de energía es el articulo:Using Hybrid Machine Learning Methods to Predict and Improve the Energy Consumption Efficiency in Oil and Gas Fields [4]. Los autores propusieron modelos de predicción del consumo de energía en empresas de petróleo y gas. A diferencia del presente trabajo donde se evalúan los metodos de Randon Forest y XGBoost, se utilizaron modelos de máquina de vectores de soporte, regresión lineal, máquina de aprendizaje extremo y red neuronal artificial. Un ejemplo donde se aborda la comparación entre los métodos Random Forest y XGBoost es el arcitulo Effective Intrusion Detection System Using XGBoost [5], donde los autores presenta un caso de éxito en el uso de este método XGBoost comparado con el modelo Ramdom Forest. A continuación de describen las técnicas utilizadas en trabajo. Estas técnicas son: Análisis exploratorio de datos, minería de datos, modelo Ramdom Forest, modelo XGboost, optimización Bayesianan, Curva Roc, área bajo la curva ROC, matriz de confusión y IBreakdown.

2.1. Análisis exploratorio y minería de datos

El análisis exploratorio de datos es una etapa crítica en el proceso de desarrollo de modelos de predicción del consumo de gas en hornos de fundir. Esta etapa implica la exploración y comprensión profunda de los datos disponibles para identificar patrones, tendencias y características clave que puedan influir en el consumo de gas. Con la utilización de la visualización de la distribución de los datos de consumo de gas se puede identificar si existen sesgos o desviaciones significativas. También se plantea la exploración de relaciones entre el consumo de gas y otras variables relevantes, como la temperatura del metal, la carga del horno y la composición del metal a fundir.

La minería de datos es una técnica que se utiliza para descubrir patrones, relaciones y conocimiento útil a partir de grandes conjuntos de datos. En el contexto de la predicción del consumo de gas en hornos de fundir, la minería de datos se puede aplicar en la identificación de características y variables más influyentes en el consumo de gas, lo que ayuda a comprender mejor los factores clave que afectan la demanda de energía en el proceso de fundición.También es de gran ayuda agrupar datos relacionados para obtener información adicional sobre distintos comportamientos de consumo de gas en diferentes condiciones operativas. También aportaría, descubrir patrones temporales o estacionales en el consumo de gas. Lo que podría ser útil para anticipar fluctuaciones en la demanda y planificar la producción de manera más eficiente.

2.2. Modelo Random Forest

El Random Forest (Bosque Aleatorio) es un algoritmo de aprendizaje automático basado en ensamblaje que combina múltiples árboles de decisión independientes para realizar predicciones. Fue propuesto por Leo Breiman en 2001 y ha demostrado ser uno de los métodos más populares y efectivos para problemas de clasificación y regresión. Para comprender el Random Forest, es esencial entender los árboles de decisión. Un árbol de decisión es una estructura de datos en forma de árbol que divide el conjunto de datos en subconjuntos más pequeños de manera recursiva. Cada nodo interno del árbol representa una pregunta sobre una característica del conjunto de datos, y cada rama representa una respuesta a esa pregunta. Los nodos hoja representan las clases o valores de destino a predecir [6].
El proceso de construcción de un Random Forest se puede resumir en los siguientes pasos:

- Seleccionar aleatoriamente muestras con reemplazo del conjunto de entrenamiento para construir múltiples subconjuntos llamados conjuntos de entrenamiento bootstrap.
- Construir un árbol de decisión independiente para cada conjunto de entrenamiento bootstrap. En cada nodo, se selecciona aleatoriamente un subconjunto de características para realizar la partición óptima.
- Repetir el proceso anterior para construir un conjunto de árboles.

Para realizar una predicción con el Random Forest, se aplica el conjunto de árboles al conjunto de datos de prueba. En el caso de clasificación, la predicción final se obtiene mediante votación mayoritaria entre los árboles, donde cada árbol emite su voto por la clase predicha.
Ventajas del Random Forest:

- Precisión y Robustez: El Random Forest generalmente tiene una alta precisión y es menos propenso al sobreajuste en comparación con un solo árbol de decisión.
- Manejo de Datos Faltantes: Puede manejar eficientemente valores faltantes en los datos, evitando la necesidad de imputación previa.
- Identificación de Importancia de Variables: Permite medir la importancia relativa de las características en función de cómo contribuyen a la precisión de las predicciones.
- Escalabilidad: Puede ser utilizado con conjuntos de datos grandes y complejos, lo que lo hace adecuado para aplicaciones de la vida real.

Random Forest es un algoritmo que combina las predicciones de múltiples árboles de decisión para lograr una mayor precisión y robustez en la clasificación datos. Su aplicación en problemas de consumo de gas en hornos de fundir puede ayudar a mejorar la eficiencia energética y reducir los costos operativos en la industria de fundición. La naturaleza explicativa del Random Forest también permite entender mejor los factores clave que influyen en el consumo de gas en este proceso industrial.

2.3. Modelo XGBoost

XGBoost (Extreme Gradient Boosting) es un algoritmo de aprendizaje automático que ha ganado popularidad debido a su eficacia en una amplia gama de problemas de clasificación y regresión. Fue desarrollado por Tianqi Chen en 2014 y se basa en la técnica de "boosting", que combina múltiples modelos más débiles para formar un modelo más fuerte y preciso [7]. Para comprender XGBoost, es importante entender el concepto de boosting. Boosting es una técnica de ensamblaje que construye múltiples modelos de aprendizaje automático de manera secuencial, donde cada modelo se enfoca en corregir los errores cometidos por el modelo anterior. Los modelos se ponderan de acuerdo con su rendimiento y se combinan para obtener una predicción final más precisa, XGBoost utiliza un algoritmo de boosting mejorado que se basa en el descenso de gradiente (gradient descent). En lugar de construir los modelos de forma secuencial, XGBoost utiliza el descenso de gradiente para optimizar directamente la función de pérdida del modelo en cada iteración. Esto permite una convergencia más rápida y una mayor eficiencia en el entrenamiento. En XGBoost, los modelos base que se utilizan son árboles de decisión, específicamente árboles CART (Classification and Regression Trees). Cada árbol se construye para predecir el gradiente de la función de pérdida, lo que significa que el árbol intenta reducir el error residual de la predicción anterior.

Las ventajas de XGBoost son :

- Precisión: XGBoost ha demostrado un rendimiento sobresaliente en una variedad de conjuntos de datos y competiciones de aprendizaje automático.
- Eficiencia y Escalabilidad: Es altamente eficiente y escalable, lo que lo hace adecuado para grandes conjuntos de datos.
- Manejo de Datos Faltantes: Puede manejar eficientemente valores faltantes en los datos sin necesidad de preprocesamiento adicional.

XGBoost es un algoritmo de aprendizaje automático altamente efectivo basado en boosting que ha demostrado un rendimiento sobresaliente en una amplia variedad de aplicaciones. Su capacidad para mejorar la precisión y generalización del modelo, junto con su eficiencia y escalabilidad, lo convierten en una herramienta poderosa para la predicción del consumo de gas en hornos de fundir y en muchas otras áreas de la ciencia y la industria.

2.4. Optimización Bayesiana

La optimización bayesiana es una técnica avanzada que se utiliza para encontrar los mejores hiperparámetros de un modelo de aprendizaje automático de manera más eficiente y efectiva. En el contexto de Random Forest y XGBoost, la optimización bayesiana se enfoca en encontrar la combinación óptima de hiperparámetros que resulte en el mejor rendimiento [8]. La optimización bayesiana es un enfoque probabilístico para optimizar funciones complejas que son costosas de evaluar. Se basa en el teorema de Bayes

para actualizar las creencias sobre la función objetivo a medida que se recopilan datos adicionales. En lugar de evaluar exhaustivamente todas las combinaciones posibles de hiperparámetros, la optimización bayesiana utiliza un enfoque de "búsqueda inteligente"para explorar el espacio de búsqueda de manera más eficiente. se basa en la construcción de un modelo probabilístico llamado "modelo de regresión gausiano"(Gaussian Process Regression, GPR). Este modelo estima la función objetivo y su incertidumbre en función de las evaluaciones previas de la función objetivo. A partir del modelo GPR, se selecciona la siguiente combinación de hiperparámetros a evaluar utilizando una estrategia llamada .ºptimización adquisitiva". La optimización adquisitiva es una función que mide la utilidad de evaluar un conjunto particular de hiperparámetros. La función de adquisición combina la incertidumbre del modelo GPR y el conocimiento previo sobre la función objetivo para determinar qué combinación de hiperparámetros puede mejorar el rendimiento del modelo.

Ventajas de la Optimización Bayesiana:

- Eficiente: En comparación con la búsqueda exhaustiva, la optimización bayesiana requiere menos evaluaciones del modelo, lo que la hace más rápida y eficiente.
- Exploración inteligente: La optimización bayesiana utiliza la información previa para dirigir la búsqueda hacia combinaciones prometedoras de hiperparámetros.
- Adaptabilidad: A medida que se realizan más evaluaciones, el modelo GPR se actualiza para reflejar las observaciones, lo que mejora la precisión de la búsqueda.

La optimización bayesiana es una para encontrar los mejores hiperparámetros de Random Forest y XGBoost de manera más eficiente. Permite mejorar la precisión y rendimiento del modelo, lo que puede llevar a un mejor desempeño en la predicción del consumo de gas en hornos de fundir.

2.5. Curva ROC, AUC-ROC y Matriz de Confusión

En de la evaluación de modelos de clasificación binaria, la Matriz de Confusión, la Curva ROC y el Área Bajo la Curva ROC (AUC-ROC) son herramientas fundamentales para analizar y comparar el rendimiento de los modelos. Estas métricas proporcionan una visión detallada del desempeño del modelo en términos de su capacidad para distinguir entre clases y clasificar correctamente las instancias [9].

La Matriz de Confusión es una tabla que resume el rendimiento de un modelo de clasificación binaria comparando las predicciones del modelo con las clases reales. Supongamos que tenemos una clasificación binaria con dos clases: clase positiva (P) y clase negativa (N). La matriz de confusión tiene cuatro elementos:

- Verdaderos Positivos (TP): El número de instancias que fueron correctamente clasificadas como positivas.
- Falsos Positivos (FP): El número de instancias que fueron incorrectamente clasificadas como positivas.
- Verdaderos Negativos (TN): El número de instancias que fueron correctamente clasificadas como negativas.
- Falsos Negativos (FN): El número de instancias que fueron incorrectamente clasificadas como negativas.

En la figura 17 se observa el esquema de una matriz de confusión.

	Modelo=1	Modelo=0
Realidad=1	Verdaderos positivos TP(A)	Falsos negativos FN (B)
Realidad=0	Falsos positivos FP(C)	Verdaderos negativos TN(D)

Figura 3: Matriz de Confusión

La Curva ROC es una representación gráfica que muestra la relación entre la Tasa de Verdaderos Positivos (True Positive Rate, TPR o Sensibilidad) y la Tasa de Falsos Positivos (False Positive Rate, FPR o Especificidad Complementaria) para diferentes valores de umbral de clasificación. En un problema de clasificación binaria, la curva ROC representa la capacidad del modelo para discriminar entre la clase positiva y la clase negativa.

- TPR (Sensibilidad o Recall): Es la proporción de instancias positivas que el modelo clasifica correctamente como positivas respecto al total de instancias positivas en el conjunto de datos. Se calcula como $TPR = TP/(TP + FN)$.
- FPR (Especificidad Complementaria): Es la proporción de instancias negativas que el modelo clasifica incorrectamente como positivas respecto al total de instancias negativas en el conjunto de datos. Se calcula como $FPR = FP/(FP + TN)$.

El Área Bajo la Curva ROC (AUC-ROC) mide el área bajo esta curva y proporciona una puntuación numérica entre 0 y 1. Un AUC-ROC mayor indica un mejor rendimiento del modelo en la clasificación de instancias positivas y negativas. En la figura 4 se representa el esquema de la curva ROC.

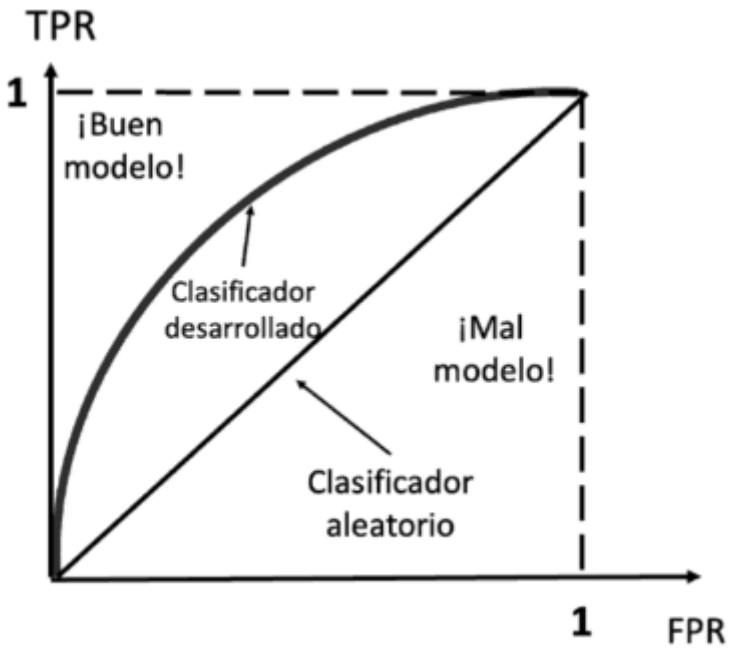

Figura 4: Curva ROC

2.6. Ibreakdown

IBreakDown es una técnica de explicabilidad de modelos de aprendizaje automático basada en el método BreakDown [10]. El método BreakDown es una técnica de explicabilidad que se enfoca en descomponer una predicción realizada por un modelo complejo en la contribución individual de cada característica presente en la instancia. Proporciona una explicación local de la predicción para una instancia específica, resaltando cómo cada característica influye en el resultado.A diferencia de Breakdown, iBreakDown tiene en cuenta la interacción entre las variables al proporcionar una explicación más detallada y precisa de cómo las características individuales de una instancia específica contribuyen a la predicción realizada por el modelo.
El proceso de iBreakDown implica los siguientes pasos:

- Seleccionar una instancia específica para la cual se desea obtener la explicación.
- Calcular las contribuciones individuales de cada característica, teniendo en cuenta las interacciones con las demás características presentes en la instancia.
- Visualizar las contribuciones considerando las interacciones, proporcionando una explicación más detallada y precisa.

Ventajas de IbreakDown

- Explicaciones más precisas: Al considerar las interacciones entre variables, iBreakDown proporciona explicaciones más precisas y realistas de cómo las características contribuyen a la predicción del modelo.
- Mayor comprensión: La explicación detallada de las interacciones permite una mayor comprensión de cómo se toman las decisiones del modelo y cómo afectan las relaciones entre las características.
- Validación de interacciones: Permite identificar interacciones significativas entre variables, lo que puede ser útil para validar y depurar el modelo.

IBreakDown es una técnica de explicabilidad que mejora el método BreakDown al considerar las interacciones entre las variables, proporcionando una explicación más precisa y detallada de cómo las características influyen en las predicciones del modelo. Esto permite una mayor comprensión del modelo y una interpretación más informada de sus resultados.

3. Metodología

3.1. Presentación y descripción de los datos utilizados

El dataset que se utilizó es sobre el proceso de fusión de aluminio del sector fundición de la Empresa Aluar [3]. El sector cuenta con dos estaciones de fundición: Estacion 1 (centro de costo 2011) y estación 2 (centro de costo 2021). Para el análisis de obtuvieron los valores desde el 1 de abril de 2021 hasta el 1 de abril de 2023. En total se analizaron 3116 registros de ordenes de fabricación. La variable a predecir es si un proceso cumple o no con el objetivo de 130 metros cúbicos por toneladas. Para comprender de mejor manera el problema se realizó un diagrama del proceso y entender de donde se obtuvieron cada una de las variables. Como se observa en la figura 5 dentro del proceso de fundición existen subprocesos. Los subproceso son planificación y programación, preparación de la carga, fusión, colada y salida del producto. El subproceso más importan para la variable en análisis es la fusión porque es donde mas se consume gas. Pero no quiere decir que los otros subprocesos impactante significativamente en el consumo. Por lo tanto se analizaran variables de todos los subprocesos.
Como se mencionó, la variable a predecir del problema es una variable categórica que indica si cumple o no un determinado proceso con el objetivo de consumo de gas. En la Figura 6 se observa la distribución del consumo de gas por tonelada. Para obtener la variable objetivo se dividieron en dos la variable consumo de gas por tonelada. Para valores menores o iguales al valor objetivo se colocó un Si y para los valores mayores se colocó No. En la figura 7 se observa el diagrama boxplot del consumo de gas por toneladas separadas por las ordenes que si cumplieron con el objetivo y por las que no.

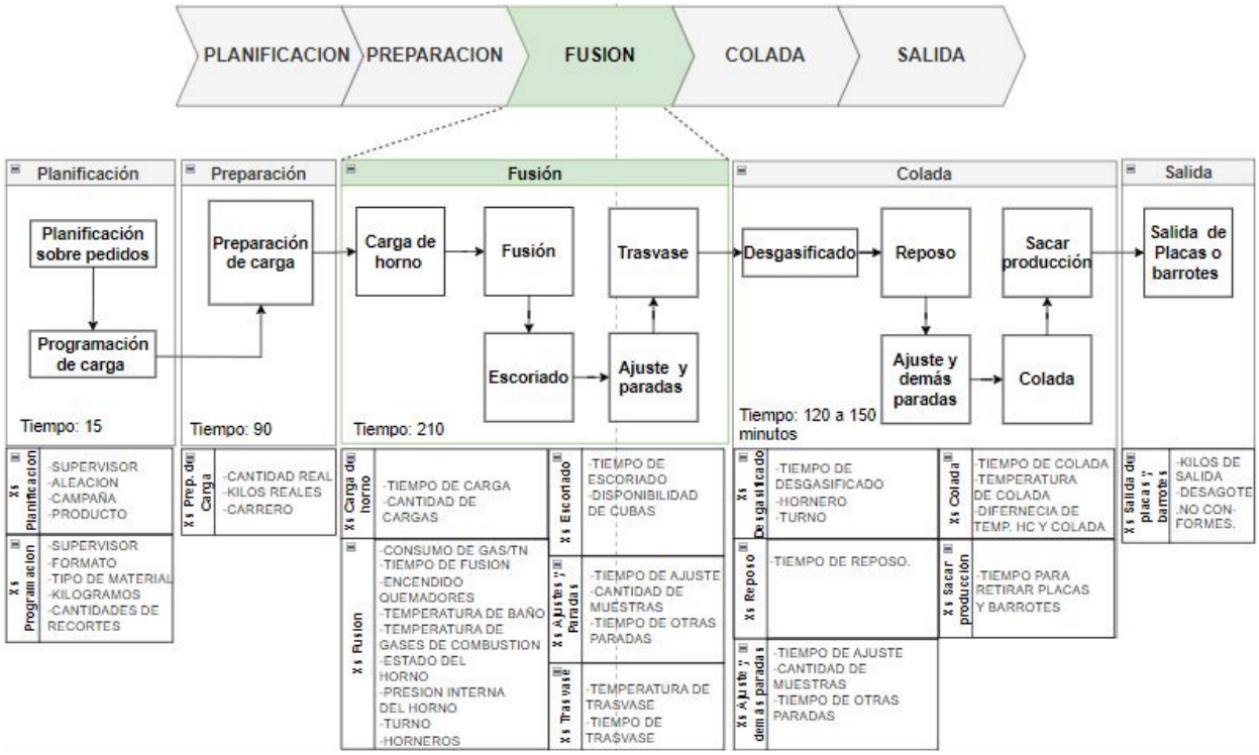

Figura 5: **:** Diagrama de proceso de fundición

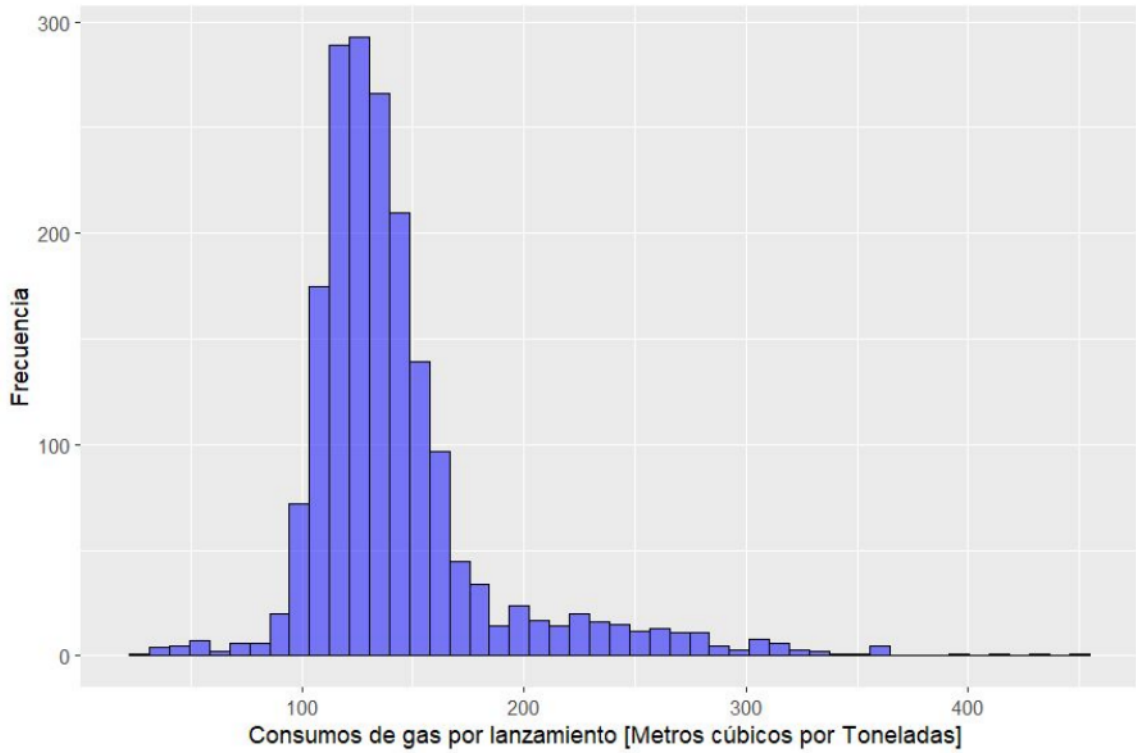

Figura 6: **:** Histograma de consumo de gas por toneladas procesadas

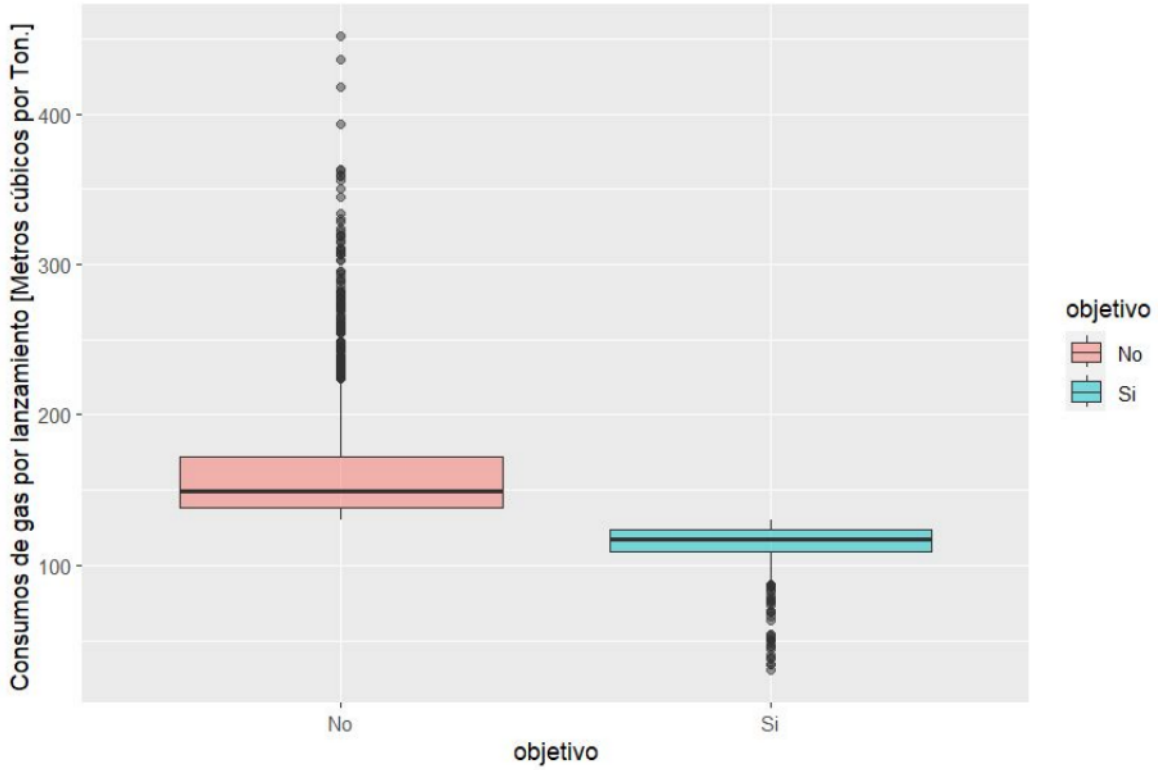

Figura 7: **:** Boxplot de variable objetivo: consumo de gas por toneladas procesadas

Las variables del proceso se obtuvieron de distintas fuentes. En total se utilizaron 7 dataset: Proceso (Tabla 1) Paradas (Tabla 2), Carga (Tabla 3), Mantenimiento (Tabla 4), Composición Química (Tabla 5, Dotación (Tabla 5) y Parámetros hornos (Tabla 7. En las tablas se detallas las variables de los dataset y la cantidad de registros. En la figura 8 se observa la relación entre los dataset.

Cuadro 1: Dataset Proceso. Total de registros 3116

Variable	**Tipo**	**Descripción**
Orden de Fabricación	Discreta	Indica el orden en el cual se va produciendo
Fecha desde	Continua	Indica la fecha y hora cuando se comenzó el proceso
Fecha hasta	Continua	Indica la fecha y hora cuando se finalizó el proceso
Centro de costo	Categórica	Indica la estación en la cual se realizó el proceso

Cuadro 2: Dataset Paradas. Total de registros 16405

Variable	**Tipo**	**Descripción**
Orden de Fabricación	Discreta	Indica el orden en el cual se va produciendo
Fecha desde	Continua	Indica la fecha y hora cuando se comenzó la parada
Fecha hasta	Continua	Indica la fecha y hora cuando se finalizó la parada
Cod. Parada	Categórica	Indica el tipo de parada
Detalle	Categórica	Detalle del código de parada

Cuadro 3: Dataset Carga. Total de registros 52602

Variable	Tipo	Descripción
Id Carga	Discreta	Código único de carga
Orden de Fabricación	Discreta	Indica el orden en el cual se va produciendo
Producto	Categórica	Descripción breve del producto
Tipo de producto	Categórica	Indica el tipo de producto
Cantidad	Discreta	Cantidad de material por producto
Contenedor	Categórica	Tipo de contenedor
kilos	Continua	Peso de la carga

Cuadro 4: Dataset Mantenimiento. Total de registros 90

Variable	Tipo	Descripción
Aviso	Discreta	Numero de orden de trabajo
Fecha	Continua	Fecha en la cual se realizó la tarea
Centro de costo	Categórica	Indica la estación en la cual se realizó el proceso
Tarea	Categórica	Tipo de tarea de mantenimiento

Cuadro 5: Dataset Composición Química. Total de registros 17373

Variable	Tipo	Descripción
Orden de Fabricación	Discreta	Indica el orden en el cual se va produciendo
Tipo de muestra	Continua	F: del horno de fundir y C: del horno de mantenimiento
Aleación	Categórica	Describe el tipo de aluminio a fabricar.

Cuadro 6: Dataset Dotación. Total de registros 2194

Variable	Tipo	Descripción
Fecha	Continua	Fecha de registro de turno
Fecha desde	Continua	Fecha de inicio del turno
Fecha hasta	Continua	Fecha de finalización del turno
Operario	Categórica	Hornero que cubre el turno
Legajo	Categórica	Número de legajo del operario
Turno	Categórica	Mañana , Tarde o Noche
Centro de costo	Categórica	Indica la estación en la cual se realizó el proceso

Cuadro 7: Dataset Parámetros hornos. Cantidad de registros 1800340

Variable	Tipo	Descripción
Fecha	Continua	Fecha del registro de la variable
Temp. Baño Est.1	Continua	Temperatura de Baño liquido del horno de fundir 1
Temp. Baño Est.2	Continua	Temperatura de Baño liquido del horno de fundir 2
Temp. Boveda Est.1	Continua	Temperatura de paredes del horno de fundir 1
Temp. Boveda Est.2	Continua	Temperatura de paredes del horno de fundir 2
Consumo de Gas Est. 1	Continua	Consumo de gas en metros cúbicos en horno de fundir 1
Consumo de Gas Est. 2	Continua	Consumo de gas en metros cúbicos en horno de fundir 2

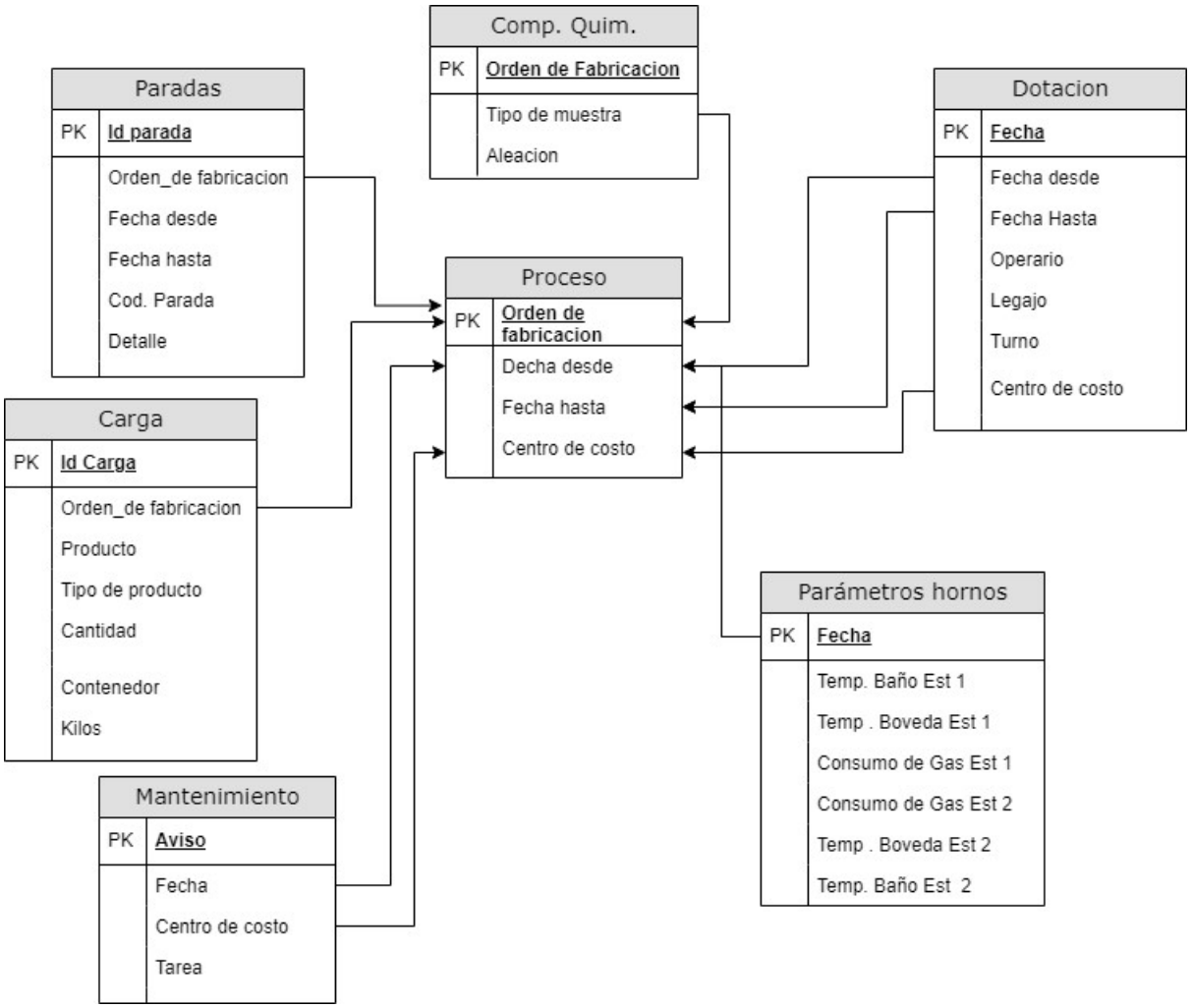

Figura 8: **:** Relación entre bases de datos

3.2. Pre-procesamiento y limpieza de los datos

En el presente trabajo para el análisis exploratorio de datos, minería de datos y limpieza se utilizaron las librerías Tidyverse [11] y data.table de R [12]. También se utilizó la librería Pandas de Python. Tidyverse es un conjunto de paquetes de software de código abierto para el lenguaje de programación R, diseñados para facilitar el análisis y la manipulación de datos. Data.table es un paquete de R diseñado para la manipulación eficiente de datos en memoria. Se caracteriza por su alta velocidad y capacidad para

trabajar con grandes conjuntos de datos.Pandas es una biblioteca de Python ampliamente utilizada para la manipulación y análisis de datos.

Las variables que se cuentan para el análisis se encontraron completas. Se eliminaron antes del análisis los procesos que no contaban con la información por falta de registros. Ya sea porque el adquisidor de datos del horno no registró los datos de un proceso determinado o por un error en la carga de los partes del proceso. A partir del procesamiento y relación entre variables se obtuvieron nuevas variables. Al agrupar la carga por orden de fabricación se obtuvo los kilos totalales. A parte se obtuvo los kilos por cada tipo de carga y su porcentual respecto al total de cada orden. También se agrupo por la variable cantidad los contenedores de carga por orden de Fabricación y el total. Luego se unieron las tablas de carga agrupada y la tabla de proceso por la variable orden de fabricación. En el caso del dataset paradas se agrupo por tipo de parada y se sumaron la duración por orden de Fabricación. A continuación se unió con la tabla de proceso por la variable orden de fabricación. Para la tabla Mantenimiento se comparo la fecha de mantenimiento para cada tipo de tarea con la fecha más cercana del proceso. A ese proceso se le asignó un cero y se contó la cantidad de procesos hasta la próxima fecha de mantenimiento por tipo. Donde nuevamente se le asigna un cero y se volvió a contar los procesos. Esta variable se obtiene para saber cuantos procesos pasaron desde la ultima intervención de mantenimiento por tipo de tarea. En el caso del dataset de composición química se agrupo por orden de Fabricación y se contó la cantidad de muestras de tipo F que corresponden con el horno de fundir. Luego se unió con la tabla proceso. En la tabla de Dotación, se obtuvo el turno, si hubo cambio de turno y el hornero en cada orden al relacionar la fecha desde y hasta de esta tabla con la tabla de proceso. En el caso de la tabla de parámetros hornos se obtuvo la temperatura máxima de baño y de bóveda de cada horno por orden de fabricación al relacionar esta tabla con la tabla proceso con la fecha del registro y las fecha desde y hasta de las ordenes. Para obtener el consumo de gas de cada orden se comparó la fecha de consumo de gas de la tabla de parámetros de horno con la fecha desde y hasta de proceso. Luego se obtuvo el consumo a partir de la diferencia de consumo entre la fecha hasta y desde. Para obtener los metros cúbicos por tonelada se dividió el consumo de gas del proceso por las toneladas procedas. Para obtener el tiempo de cada proceso se realizó la resta entre fecha desde y hasta. Par obtener el tiempo de quemadores encendidos, al tiempo de proceso se le resto el tiempo total de paradas del proceso. Para obtener el tiempo neto, al tiempo total se le resto las parada programadas. A partir de las fechas de los procesos de obtuvieron el día de la semana, la semana del año y el mes. Por último a partir de las variables categóricas de obtuvieron dummies para utilizarlas en los métodos predictivos Random Forest y XGBoost.

3.3. Análisis exploratorio de datos

Para el análisis exploratorio de datos se utilizaron histogramas para las variables numéricas más significativas y boxplot en el caso de las variables categóricas más significativas. También se utilizaron gráficos de scatterplot de las variables significativas con respecto al consumo de gas. También para analizar la correlación entre las variables se realizó una matriz de correlación. En la figura 9 se aprecia que las toneladas de carga se concentran en aproximadamente 9 toneladas de una capacidad total de horno de 12 toneladas. Con respecto a las variables P_Programada y tiempo se aprecia valores atípicos a la derecha del gráfico. Por lo tanto hay procesos que se extienden. El tiempo de quemadores se centra en 2.3 horas. Con respecto a las tares que se realizan en los quemadores se aprecia que en muchos casos pasan demasiado tiempo entre intervenciones. En el caso de la temperatura de baño se aprecia que hay casos donde el horno se pasa de la temperatura estándar de menos de 760 ^{o}C. En la figura 10 se aprecia que cuanto más toneladas contenga la carga menos gas de consumo. También se observa que cuando aumenta el tiempo,las paradas programadas, el tiempo de los quemadores encendidos y la temperatura de baño aumenta el consumo de gas. Por otro lada no se aprecia grandes diferencia del consumo de gas con respecto a la intervención de quemadores. Al analizar la figura 11 se aprecia que los turnos que más consumen gas con la combinación del Turno mañana-noche, Tarde-maña y Noche-Tarde. Esto se debe a que se salteó un turno de trabajo donde el horno estuvo parado. Y se consumió más gas para poder calentar el horno.En la figura 12 se aprecia que a medida que hay más productos de la Planta de Puerto Madryn en la carga (Lingotes de aluminio puro) se consume menos cantidad de gas por toneladas procesadas. En la figura 13 se observa que no hay grandes variaciones en el consumo de gas si dentro del proceso existe n cambio de turno o no. En la figura 14 se aprecia la correlación entre las variables. Lo que se observa es que las variables que más correlación tienen con el consumo de gas son el tiempo entre procesos, las parada programadas, las toneladas, la utilización de lingotes y la temperatura del baño liquido.

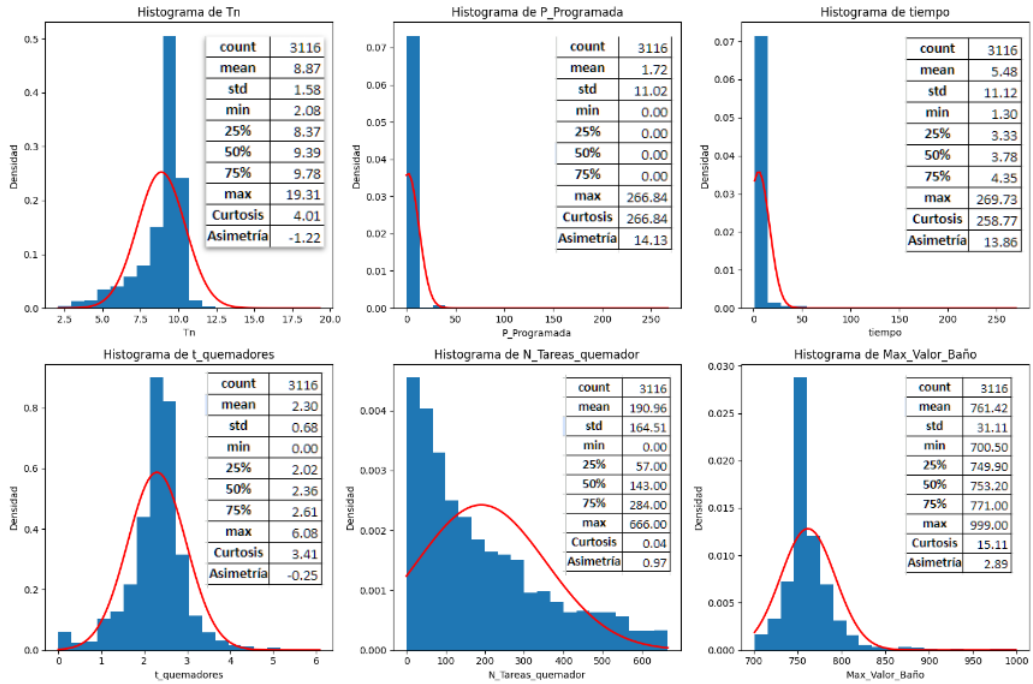

Figura 9: Histograma con descripción estadística de algunas de las variables más significativas. Tn: toneladas de carga, P_Programada:duración de la parada, Tiempo: duración del proceso, T_quemadores: tiempo de quemadores encendidos, N_Tareas_quemadores: Cantidad de ordenes de fabricación que pasaron desde la última intervención de los quemadores y Max_Valor_Baño: máxima temperatura del baño liquido

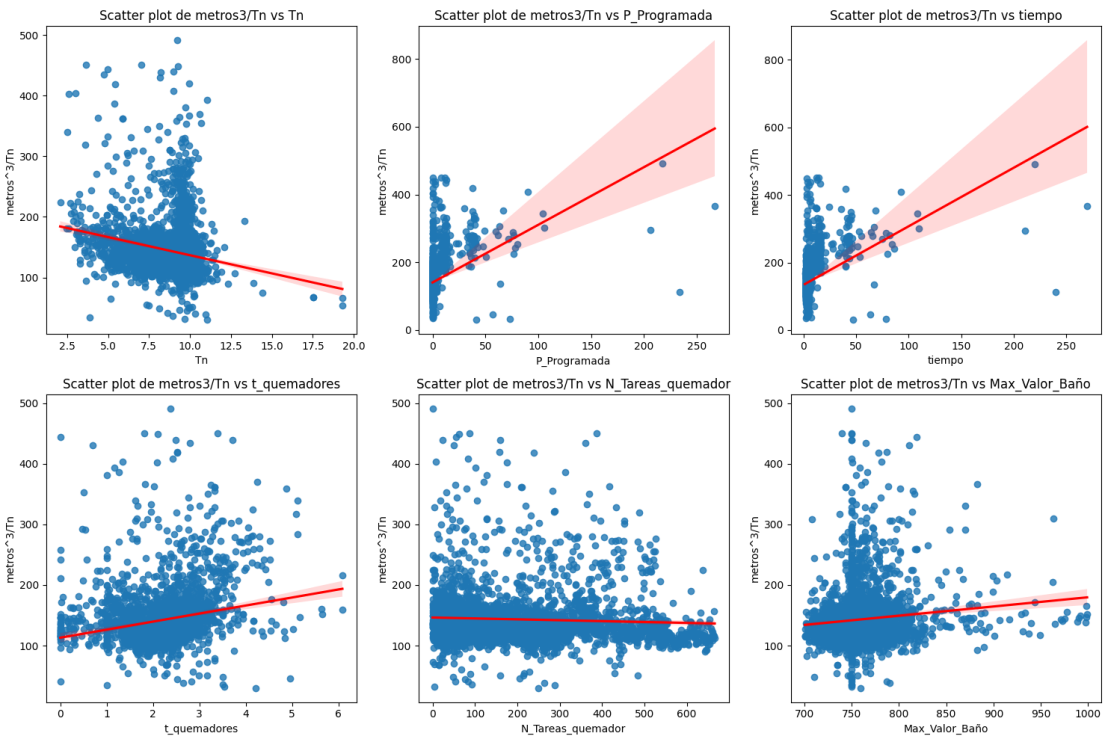

Figura 10: Scatterplot con linea de tendencia de las variables más significativas

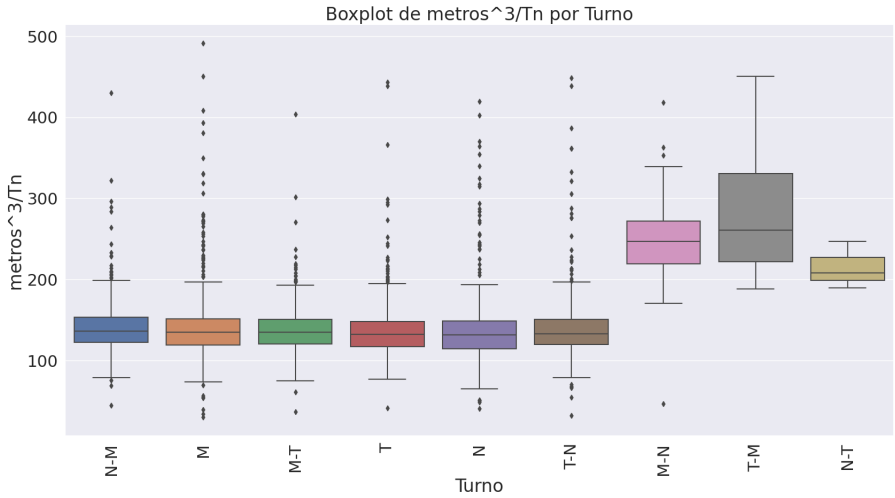

Figura 11: Boxplot de los diferentes turnos. M: mañana, T: tarde y N:Noche

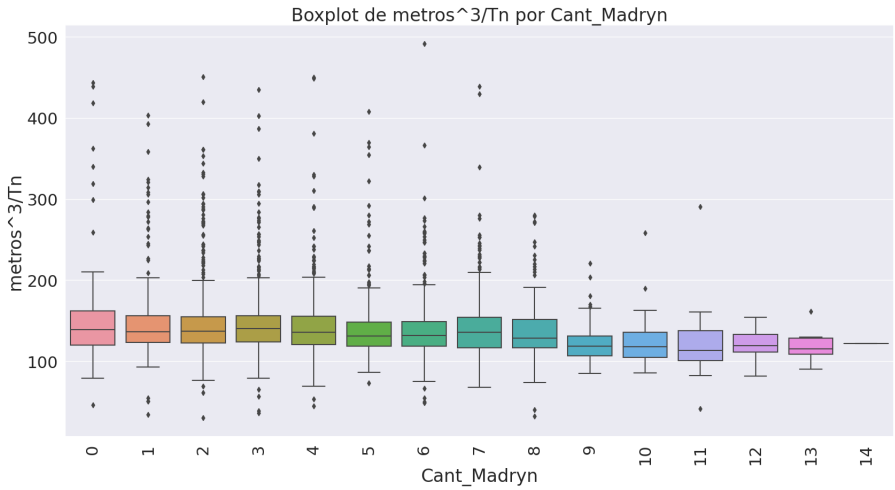

Figura 12: Boxplot de la cantidad de productos provenientes de Madryn en la carga

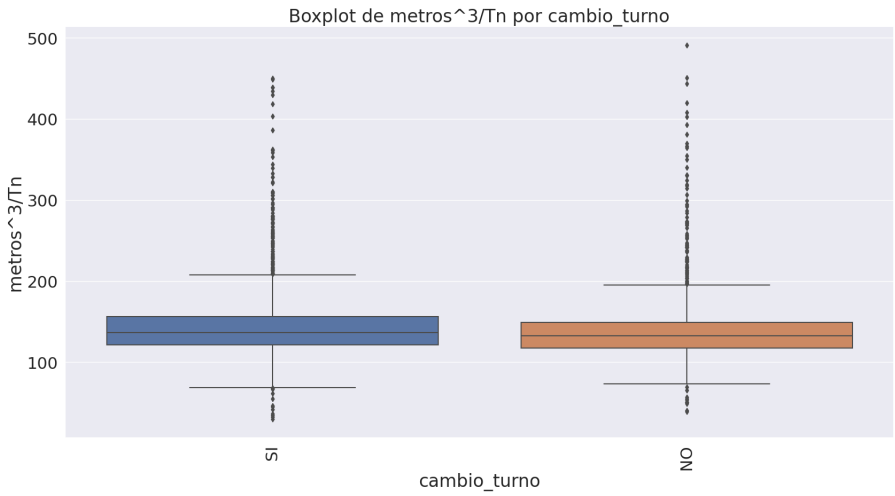

Figura 13: Boxplot de si existe cambio de turno dentro del proceso

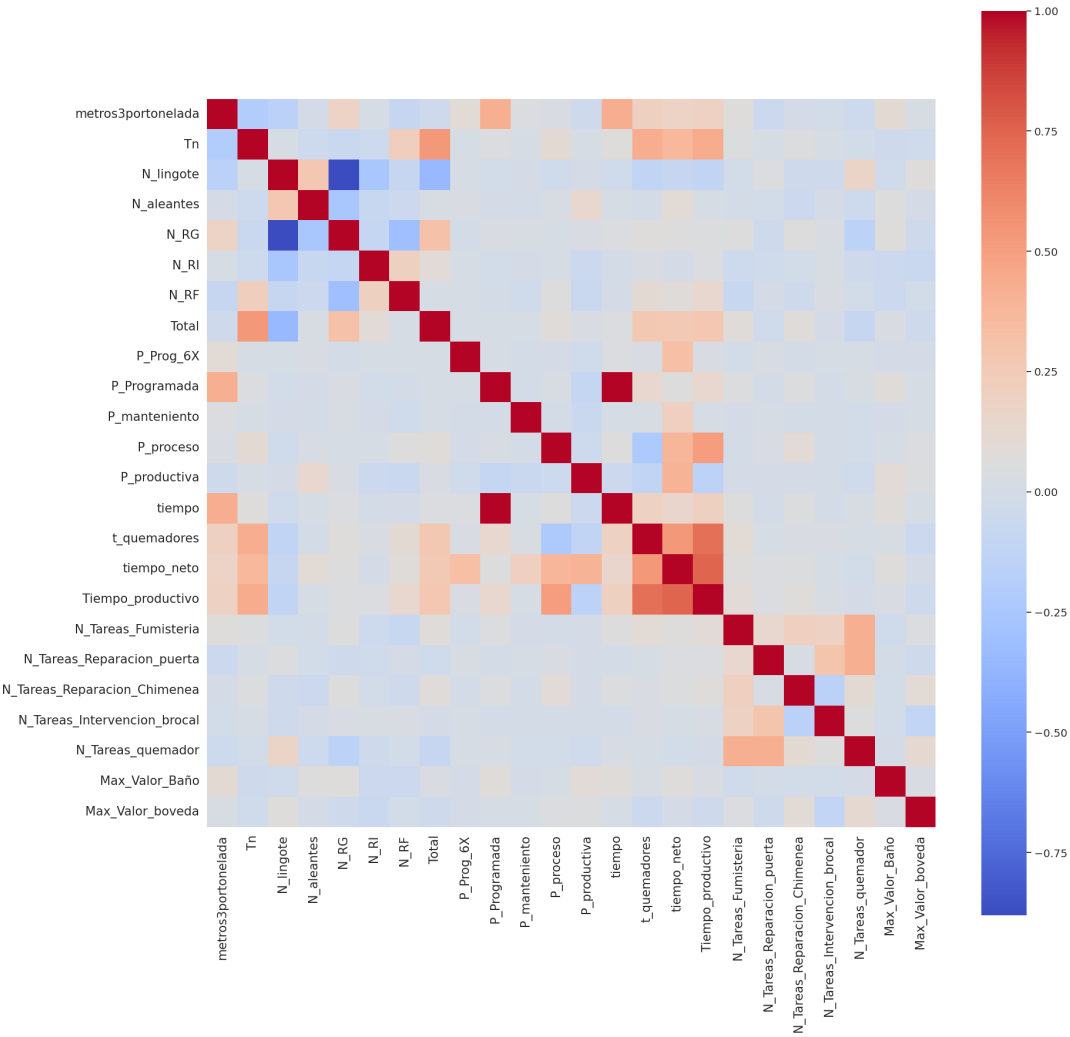

Figura 14: Matriz de correlación de variables numéricas

3.4. Descripción de las técnicas de análisis de modelado

Una vez procesados los datos se realizó la partición de los datos en entrenamiento y test. Para realizar realizar la partición se utilizó la función *train_ test_ split* en Python. La partición que se realizó fue 80 % de los datos para el entrenamiento y 20 % para testeo. Se utilizo una semilla para poder tener replicabilidad en los resultados. También la partición se realizó de forma estratificado respecto a la variable a predecir. Para realizar la predicción se utilizo los algoritmo Random Forest y XGBoost [13]. Lo primero que se realizó fue una optimización bayesiana para los dos algoritmos para obtener los mejores hiper parámetros. La optimización bayesiana permite encontrar los mejores valores para los hiper parámetros al combinar la información previa con las evaluaciones obtenidas durante el proceso de optimización. Tiene la ventaja de ser eficiente en la explotación del espacio de hiper parámetros. [8]. En ambos casos se utilizó validación cruzada de 5 particiones. Como métrica para comparar la mejor combinación de hiper-parámetros se utilizó el área bajo la curva ROC. La curva ROC proporciona una medida global del rendimiento del modelo de clasificación al mostrar la relación entre la tasa de verdaderos positivos (sensibilidad) y la tasa de falsos positivos (1 -especificidad) [14]. Como cantidad de iteraciones se fijo en 50 en ambos casos.
A continuación se detallan los hiper-parametros utilizados para rambom forest en la optimización Bayesiana:

- Profundidad máxima (Max Depth): Este parámetro determina la profundidad máxima de cada árbol de decisión en el bosque aleatorio. Representa el número máximo de niveles que puede tener un árbol. Una mayor profundidad máxima permite capturar relaciones más complejas en los datos, pero también aumenta el riesgo de sobreajuste.

- Muestras mínimas por hoja (Min Samples Leaf): Este parámetro especifica el número mínimo de muestras requeridas para que un nodo sea considerado una hoja. Una hoja es un nodo terminal en un árbol de decisión donde no se realizan más divisiones. Establecer un valor más alto para muestras mínimas por hoja resulta en árboles más simples, ya que evita crear nodos con muy pocas muestras, lo cual puede conducir al sobreajuste.
- Muestras mínimas por división (Min Samples Split): Este parámetro establece el número mínimo de muestras requeridas para realizar una división en un nodo interno. Si el número de muestras en un nodo es inferior a este umbral, el nodo no se dividirá y se convertirá en una hoja. Al igual que con muestras mínimas por hoja, un valor más alto para muestras mínimas por división promueve árboles más simples y reduce el riesgo de sobreajuste.
- Número de estimadores (N Estimators): Determina el número de árboles de decisión individuales que se crearán en el conjunto. Aumentar el valor de n estimators generalmente mejora el rendimiento del modelo, ya que permite que el conjunto se beneficie de árboles más diversos y captura una gama más amplia de patrones en los datos. Sin embargo, llega un punto en el que los beneficios se estabilizan y agregar más estimadores no produce mejoras significativas, mientras aumenta la complejidad computacional.

En la tabla 8 se presentan los rangos de los hiper-parámetros que se utilizaron en la optimización bayesiana para el modelo Random Forest.

Cuadro 8: Rango hipe-parámetros para Random Forest en la optimizaciòn bayesiana

Hiper-parámetro	**Valor**
max depth	[1 -30]
min samples leaf	[1 - 20]
min samples split	[2 - 30]
n estimators	[10, 500]

En la tabla 9 se Presentan los valores de los mejores hiper-parámetros del modelo Random Forest al aplicar la optimización bayesiana.

Cuadro 9: Mejores hipe-parámetros para Random Forest

Hiper-parámetro	**Valor**
max depth	22
min samples leaf	1
min samples split	2
n estimators	259

A continuación se detallan los hiper-parametros utilizados para XGBoost en la optimización Bayesiana:

- Colsample bytree: Este parámetro determina la proporción de características (columnas) que se utilizarán al construir cada árbol en el conjunto. Por lo general, se especifica como un valor entre 0 y 1.
- Tasa de aprendizaje (Learning rate) : Este parámetro controla la contribución de cada árbol al modelo final. Reduce la importancia de cada árbol mediante la multiplicación de una tasa de aprendizaje (valor entre 0 y 1) a los valores predichos por cada árbol.
- Profundidad máxim (Max depth): Este parámetro especifica la profundidad máxima de cada árbol en el conjunto. Controla la complejidad de cada árbol y su capacidad para capturar relaciones más detalladas en los datos. Valores más altos permiten árboles más profundos y más complejos, lo que puede llevar a un sobreajuste si no se controla adecuadamente.

- Número de estimadores (N estimators): Este parámetro indica la cantidad de árboles de decisión (estimadores) que se incluirán en el conjunto. Cada árbol se construye de forma secuencial, corrigiendo los errores del modelo anterior. Un número mayor de estimadores generalmente mejora la capacidad de generalización del modelo, pero también aumenta el tiempo de entrenamiento.
- Subsample: Este parámetro especifica la proporción de muestras (filas) que se utilizarán para entrenar cada árbol en el conjunto. .

En la tabla 10 se presentan los rangos de los hiper-parámetros que se utilizaron en la optimización bayesiana para el modelo Random Forest.

Cuadro 10: Rango hipe-parámetros para XGBoost en la optimización bayesiana

Hiper-parámetro	**Valor**
colsample bytree	[0.5 - 1.0]
learning rate	[0.01 - 0.1]
max depth	[1 - 30]
n estimators	[10 - 500]
subsample	[0.5 - 1.0]

En la tabla 11 se Presentan los valores de los mejores hiper-parámetros del modelo XGboost al aplicar la optimización bayesiana.

Cuadro 11: Mejores hipe-parámetros para XGBoost

Hiper-parámetro	**Valor**
colsample bytree	0.6580
learning rate	0.0376
max depth	28
n estimators	423
subsample	0.4945

Una vez obtenidos los mejores hiper parámetros se entrenaron los modelos Random Forest y XGBoost con los datos de entrenamiento. Con el modelo entrenado se predijo los valores de test. A continuación se evaluó el modelo utilizando la métrica del área bajo la curva roc. En la curva roc se compararon los valores de la predicción con los valores de testeo. Se compararon los rendimientos de los dos modelos a partir de los resultados de la curva roc. También para comparar los modelos utilizados se utilizó la matriz de confusión.

Por ultimo, para agregarle explicación al modelo de caja Negra XGboost, que fue el que mejor resultado obtuvo, se utilizó el algoritmo iBreakdown. IBreakdown se basa en el concepto de descomposición de las predicciones del modelo en términos de contribuciones individuales de las variables utilizadas en el modelo. Esta descomposición permite entender cómo cada variable contribuye a la predicción final y cómo interactúan entre sí[10]. Este método se aplica de a un valor de test
El algoritmo lo que hace es:

- Se toma de a una variable de Test y se la remplaza en todos los registros de Testeo y se obtiene la predicción. Por ejemplo si el valor elegido de test para centro de costo es 2011, se reemplaza todos los registros de entrenamiento por 2011 en centro de costo y se obtiene la predicción
- Para todas las variables se obtiene la diferencia entre la predicción del punto anterior y el promedio de las predicciones de test. Como iBreakdown también tiene en cuenta las iteraciones entre las variables también se toman las variables de a pares y se obtiene el valor de la predicción y se obtiene la diferencia respecto al valor promedio de test. Si la diferencia es mayor en la iteración que en las variables individuales y la sumas de las individuales, se toma en cuenta la iteración y se descartan las individuales. Caso contrario se toman las individuales. Una vez obtenidas las diferencias, se comparan las variables y se las ordena de forma decreciente

- Una vez que se encuentran ordenadas de a una se van adicionando de forma decreciente. Para la primer variable, el aporte es la diferencia entre el valor de perdición al fijar la variable de test en el dataset de entrenamiento y el valor promedio de la predicción de test.
- Para las variables sucesivas se van manteniendo fijas las anteriores y se fija la nueva variable y se obtiene la predicción y se la compara con las diferencia acumuladas. Por eso se dice que es un método aditivo.
- En la última variable tomas los registros de test van a tener los valores del registro Test seleccionado por lo tanto va a tener el mismo valor de predicción.

Para analizar la explicación del modelo XGBoost en el dataset en estudio se utilizaron cuatro ejemplos de la partición Test. Para la implementación se utilizó la librería Dalex en Python.

4. Resultados y discusión

4.1. Presentación y análisis de resultados obtenidos

En la Tabla 12 se presentan los resultados obtenidos en la predicción de los modelos Random Forest y XGBoost. Se observa que XGBoost tiene mejor rendimiento respecto a Random Forest tanto en entrenamiento como en testeo.

Cuadro 12: Comparación de modelos Random Forest y XGBoost a partir del área bajo la curva roc (AUC) para entrenamiento y testeo

Modelo	AUC en Entrenamiento	AUC Testeo
Random Forest	0.911	0.905
XGBoost	0.939	0.924

En la Figura 15 se observa la curva roc para el modelo Random Forest y en la figura 16 se observa la curva roc para el modelo XGBoost. En las Figuras se compara la tasa de verdaderos positivos respecto a la tasa de falsos positivos. También se muestra el valor del área bajo la curva. Cuando la curva esta más alejada a la linea punteado mejor es la predicción del modelo.

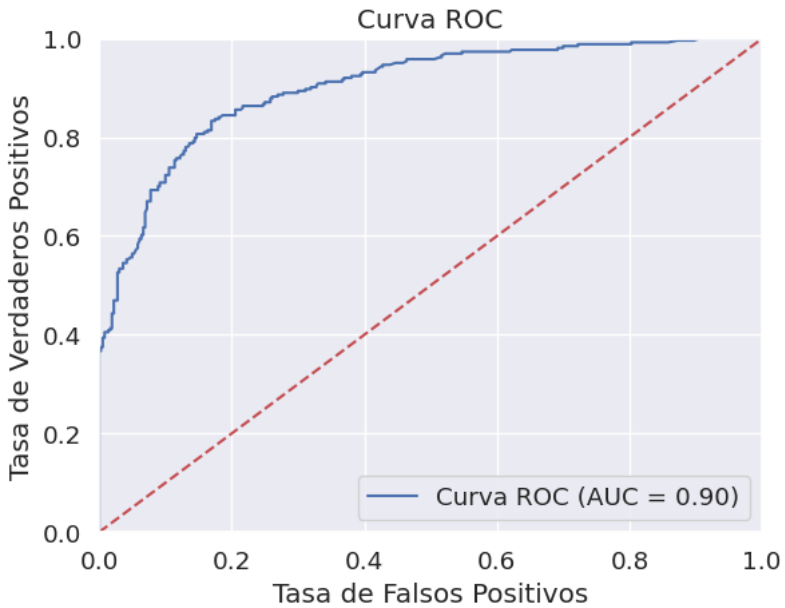

Figura 15: Curva roc del modelo Random Forest

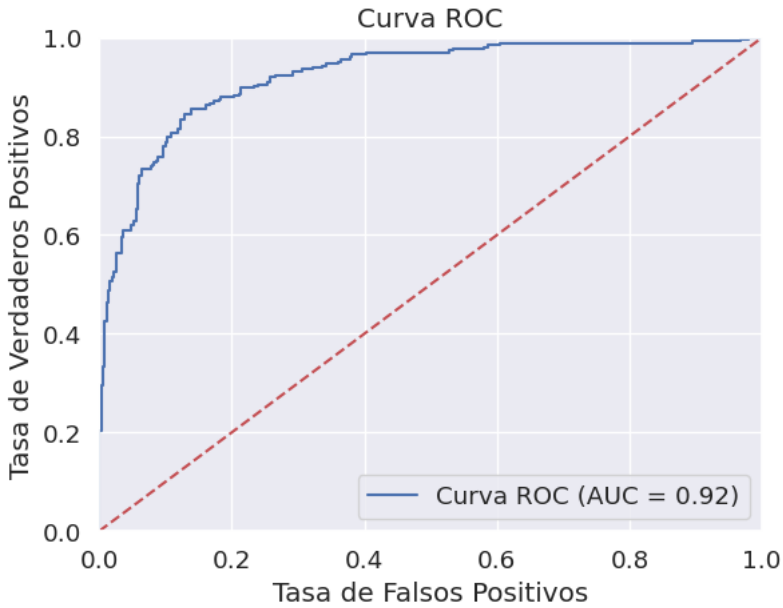

Figura 16: Curva roc del modelo XGBoost

En la figura 17 se presenta la matriz de confusión para el modelo Ramdom Forest y el modelo XGBoost. Lo que se observa que el modelo Ramdom forest se confunde menos en las ordenes de fabricación que no cumplen con el objetivo que el modelo XGBoost. También que el modelo XGboost se confunde menos en las ordenes de fabricación que cumplen con el objetivo respecto al modelo Random Forest.

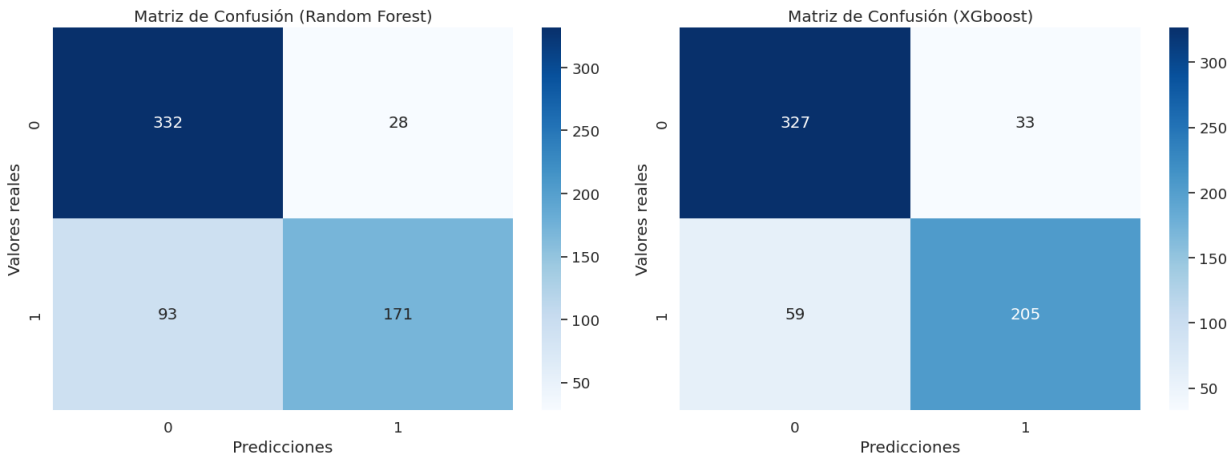

Figura 17: Matriz de confusión para el modelo Random Forest y XGBoost. El valor cero corresponde a las ordenes de Fabricación que no cumplen con el objetivo de 130 metros cúbicos por toneladas. El valor uno corresponden a los que cumplen.

En la Figuras 18 Se observan un valor de testeo donde se observan cuales son las variables que más impactan en la predicción. En este caso los kilogramos de la carga es lo que mas impacta positivamente. En segundo lugar la semana del año en forma negativa. En tercer lugar el la temperatura del baño liquido en forma negativa.En la figura 20 se aprecia otro caso donde en este caso la variable que mas impactan es las tareas de mantenimiento en el quemador, que pertenezca al horno de fundir 1 y las reparaciones en las puertas. Las tres variables en forma negativa. En la figura 20 se observa que las variables que mas impactan en forma positiva son la iteración entre el tiempo de los quemadores y el tiempo entre procesos, las tareas en quemadores y los kilos cargados. En la figura 21 se aprecia que el tiempo , el tiempo de los quemadores encendidos y las reparaciones de chimenea son las variables que más impactan de forma

negativa.

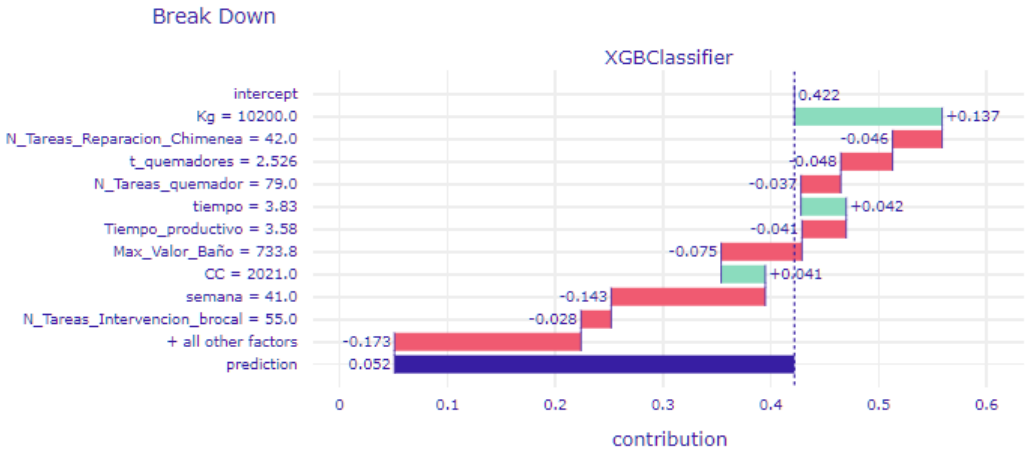

Figura 18: Aplicación de IBreakDown en un registro [0] del dataset testeo

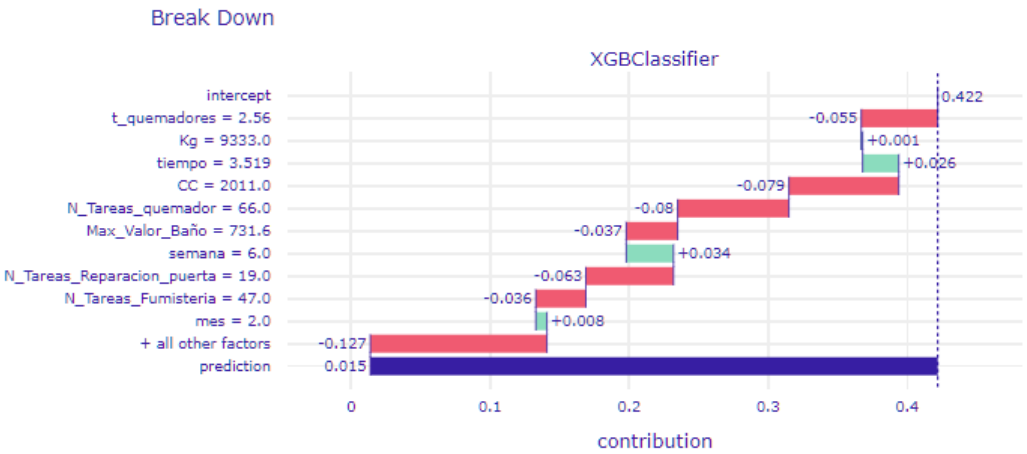

Figura 19: Aplicación de IBreakDown en un registro [25] del dataset testeo

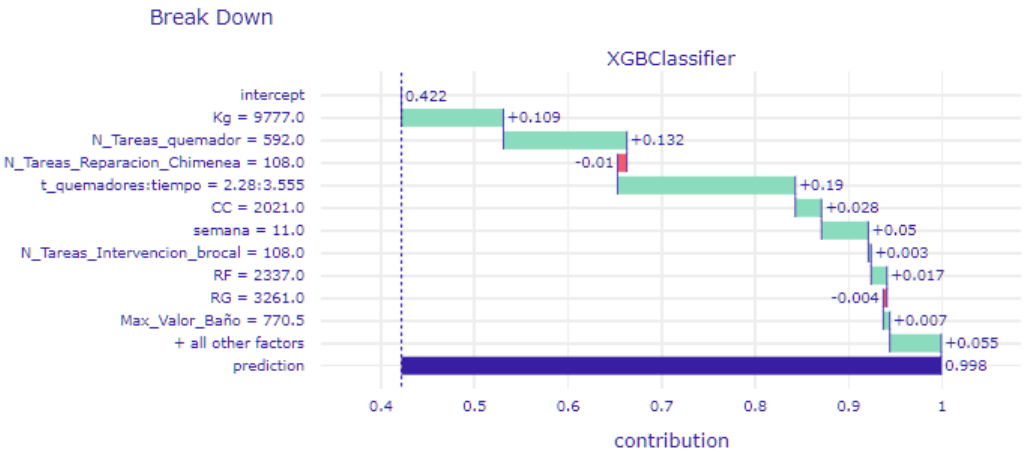

Figura 20: Aplicación de IBreakDown en un registro [50] del dataset testeo

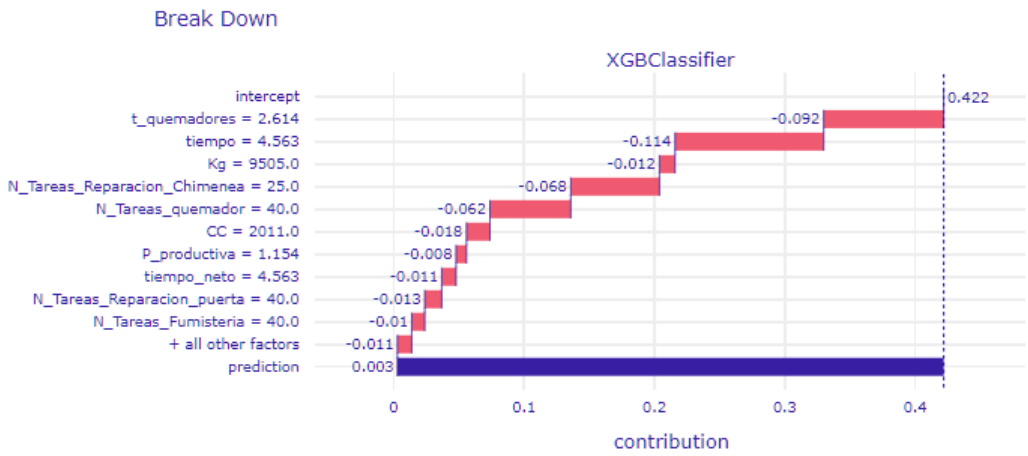

Figura 21: Aplicación de IBreakDown en un registro [200] del dataset testeo

4.2. Discusión de los resultados y su relevancia

A partir de la observación de los resultados, se destaca que el algoritmo XGBoost presenta el mayor rendimiento al comparar los valores del área bajo la curva ROC en los modelos presentados. Sin embargo, es importante mencionar que para los procesos que no cumplen el objetivo de consumo de gas, el modelo que demostró mejor desempeño fue Random Forest. Por otro lado, para los procesos que sí cumplen el objetivo de consumo de gas, el modelo más efectivo resultó ser XGBoost.

Asimismo, se encontró que IbreakDown es una herramienta valiosa para agregar explicatividad a modelos de caja negra como XGBoost. A través de los ejemplos presentados, se pudo observar que las variables más influyentes en el consumo de gas son: kilogramos de carga, tiempo neto, tiempo de quemadores, ubicación del horno, temperatura del baño líquido, y el mantenimiento de quemadores, puertas y chimeneas.

4.3. Limitaciones y posibles mejoras

Durante el desarrollo del trabajo, se encontraron ciertas limitaciones que afectaron el análisis. En primer lugar, la cantidad de datos en el dataset de procesos fue reducida, con tan solo 3116 registros de órdenes de fabricación disponibles. Esta limitación se debió a la eliminación de algunos procesos debido a problemas de comunicación con el adquisidor de datos y errores en la carga. Otra limitación importante fue la falta de variables adicionales relacionadas con los parámetros de los hornos, que podrían haber sido de gran utilidad para optimizar su rendimiento. Algunas de estas variables ausentes incluyen la sonda lambda en la chimenea del horno para medir la calidad de los humos de escape, así como la temperatura de la chimenea y la presión interna del horno. Además, cabe destacar que el uso de Ibreakdown se limita a proporcionar explicaciones para un solo registro a la vez, lo que no permite obtener explicatividad para todos los registros de testeo.

Para mejorar la precisión de la predicción, se proponen diversas acciones. En primer lugar, se considera fundamental adquirir una mayor cantidad de datos mediante una revisión más amplia en un rango de tiempo más extenso. Para evitar la pérdida de información valiosa, se sugiere la instalación de dispositivos como UPS (Sistemas de Alimentación Ininterrumpida) para asegurar la continuidad del registro de datos. Asimismo, se recomienda agregar nuevas variables importantes para la eficiencia del horno, como la sonda lambda y la presión interna del mismo. Estos parámetros adicionales podrían brindar una mayor comprensión del proceso y mejorar la calidad de la predicción. Otra opción para mejorar la predicción es probar diferentes modelos de machine learning, como LightGBM o redes neuronales, que podrían proporcionar resultados más precisos y complementarios a los modelos actuales. También se sugiere la posibilidad de realizar una hibridación entre los modelos presentados para evaluar si esta estrategia conduce a una mejora en la predicción general.

5. Conclusión

En el presente informe, se ha abordado el objetivo de predecir si un proceso va a consumir menos de 130 metros cúbicos por tonelada, utilizando dos métodos de aprendizaje automático: Random Forest y XGBoost. Además, se ha identificado el impacto de diferentes variables en el consumo de gas del proceso mediante el uso de la herramienta IBreakdown. Los hallazgos principales destacan que el modelo XGBoost ha mostrado el mayor rendimiento al comparar los valores del área bajo la curva ROC en los diferentes modelos presentados. Sin embargo, es importante mencionar que, para aquellos procesos que no cumplen el objetivo de consumo de gas, el modelo Random Forest demostró un mejor desempeño en las predicciones. Uno de los aspectos más valiosos de este trabajo ha sido la utilización de la herramienta IBreakdown, la cual ha permitido agregar explicatividad a los modelos de caja negra como XGBoost. Gracias a esto, se han identificado las variables más influyentes en el consumo de gas, entre las que se encuentran: kilogramos de carga, tiempo neto, tiempo de quemadores, ubicación del horno, temperatura del baño líquido y el mantenimiento de quemadores, puertas y chimeneas. Esta información es crucial para conocer las áreas en las que se pueden realizar modificaciones para reducir la cantidad de gas consumido, y así mejorar la eficiencia del proceso. Por lo tanto, se recomienda tomar las siguientes acciones para bajar el consumo de gas. En primer trabajar en lo posible con el horno totalmente lleno. Analizar los procesos en los cuales se extiende el tiempo respecto a la media. Analizar porque tienen menor consumo de gas el horno 2 que el 1. Respectar la frecuencia de mantenimiento de los hornos, tanto de la parte refractaria, como el aislamiento de tapas , puertas y chimenea. Por otro lado se recomienda instalar una termocupla de medición continua de baño liquido para mejorar dicho registro y automatizar los tiempos de los procesos.

No obstante, durante el desarrollo del trabajo, se encontraron ciertas limitaciones que afectaron el análisis. La cantidad de datos en el dataset de procesos fue reducida, lo cual limitó la precisión de las predicciones. Asimismo, la falta de variables adicionales relacionadas con los parámetros de los hornos también ha sido una limitante importante. Para mejorar la precisión de las predicciones en futuros trabajos, se proponen varias acciones. En primer lugar, es fundamental adquirir una mayor cantidad de datos mediante una revisión más amplia en un rango de tiempo más extenso. Para evitar la pérdida de información valiosa, se sugiere la instalación de dispositivos como UPS para asegurar la continuidad del registro de datos. Además, se recomienda agregar nuevas variables importantes para la eficiencia del horno, como la sonda lambda en la chimenea para medir la calidad de los humos de escape, así como la temperatura y la presión interna del horno. Estos parámetros adicionales podrían brindar una mayor comprensión del proceso y, en consecuencia, mejorar la calidad de las predicciones. Por otro lado, explorar diferentes modelos de machine learning, como LightGBM o redes neuronales, también podría proporcionar resultados más precisos y complementarios a los modelos actuales.

Este informe ha proporcionado información valiosa sobre la predicción del consumo de gas en procesos industriales. Los modelos Random Forest y XGBoost han demostrado su eficacia en diferentes escenarios, y la herramienta IBreakdown ha aportado explicatividad a los modelos de caja negra, permitiendo identificar las variables más influyentes en el consumo de gas. A pesar de las limitaciones encontradas, las recomendaciones propuestas para futuros trabajos ofrecen oportunidades para mejorar la precisión de las predicciones y, en última instancia, optimizar el consumo de gas en el proceso industrial estudiado.

Bibliografía

[1] Christina Windmark et al. "Investigation on Resource-Efficient Aluminium Recycling – A State of the Art Review". En: abr. de 2022. ISBN: 9781643682686. DOI: 10.3233/ATDE220122.

[2] Stefano Capuzzi y Giulio Timelli. "Preparation and Melting of Scrap in Aluminum Recycling: A Review". En: *Metals - Open Access Metallurgy Journal* 8 (abr. de 2018), pág. 249. DOI: 10.3390/met8040249.

[3] *Aluar*. Ultima visita 23/05/2023. URL: https://www.aluar.com.ar/.

[4] Xiangyang Zhang Jun Li Yidong Guo y Zhanbao Fu. "Using Hybrid Machine Learning Methods to Predict and Improve the Energy Consumption Efficiency in Oil and Gas Fields". En: (2021). URL: https://www.hindawi.com/journals/misy/2021/5729630.

[5] Sukhpreet Singh Dhaliwal, Abdullah-Al Nahid y Robert Abbas. "Effective Intrusion Detection System Using XGBoost". En: *Information* 9.7 (2018). ISSN: 2078-2489. URL: https://www.mdpi.com/2078-2489/9/7/149.

[6] L Breiman. "Random Forests". En: *Machine Learning* 45 (oct. de 2001), págs. 5-32. DOI: 10.1023/A:1010950718922.

[7] Tianqi Chen y Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". En: *CoRR* abs/1603.02754 (2016). URL: http://arxiv.org/abs/1603.02754.

[8] Vu Nguyen. "Bayesian Optimization for Accelerating Hyper-Parameter Tuning". En: *2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*. 2019, págs. 302-305. DOI: 10.1109/AIKE.2019.00060.

[9] Pablo Belenguer Emiliano Soria. *Inteligencia artificial: Casos prácticos con aprendizaje profundo*. 2022. ISBN: 9789587924411. URL: https://books.google.com.ar/books?id=XHugEAAAQBAJ.

[10] Alicja Gosiewska y Przemyslaw Biecek. "iBreakDown: Uncertainty of Model Explanations for Non-additive Predictive Models". En: *CoRR* abs/1903.11420 (2019). arXiv: 1903.11420. URL: http://arxiv.org/abs/1903.11420.

[11] *Tidyverse*. Ultima consulta 23-06-2023. URL: https://www.tidyverse.org/.

[12] Matt Wiley Joshua F. Wiley. *Introduction to data.table*. 2020. URL: https://doi.org/10.1007/978-1-4842-5973-3_7.

[13] *XGBoost*. Ultima consulta 23-06-2023. URL: https://xgboost.ai.

[14] Tom Fawcett. "Introduction to ROC analysis". En: *Pattern Recognition Letters* 27 (jun. de 2006), págs. 861-874. DOI: 10.1016/j.patrec.2005.10.010.

A. Anexo

A.1. Código fuente utilizado

En el siguiente repositorio se puede consultar el código fuente utilizado en el trabajo:

Web: https://github.com/rensogil/Eficiencia_energetica_hornos

A.2. Ejemplos de aplicación de IBreakdown

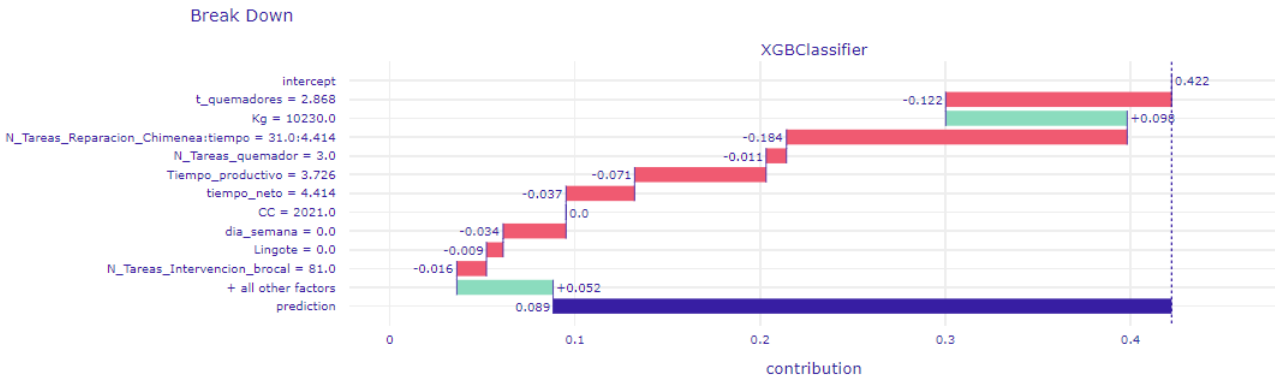

Figura 22: Aplicación de IBreakDown en un registro [250] del dataset testeo

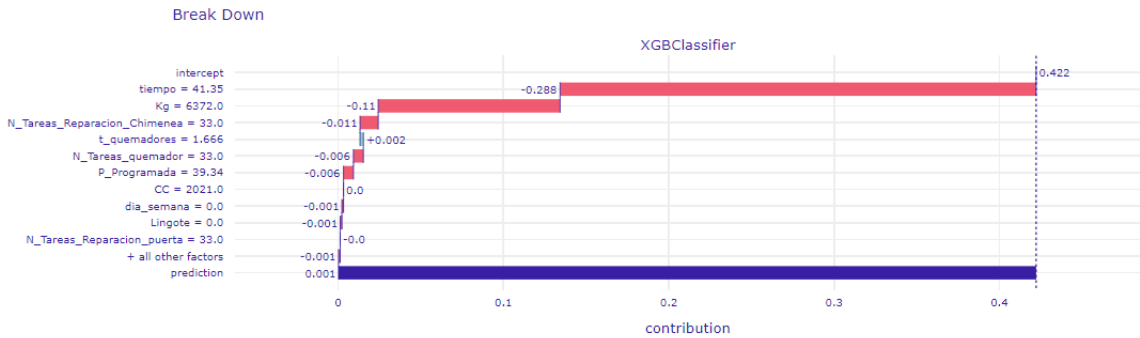

Figura 23: Aplicación de IBreakDown en un registro [300] del dataset testeo

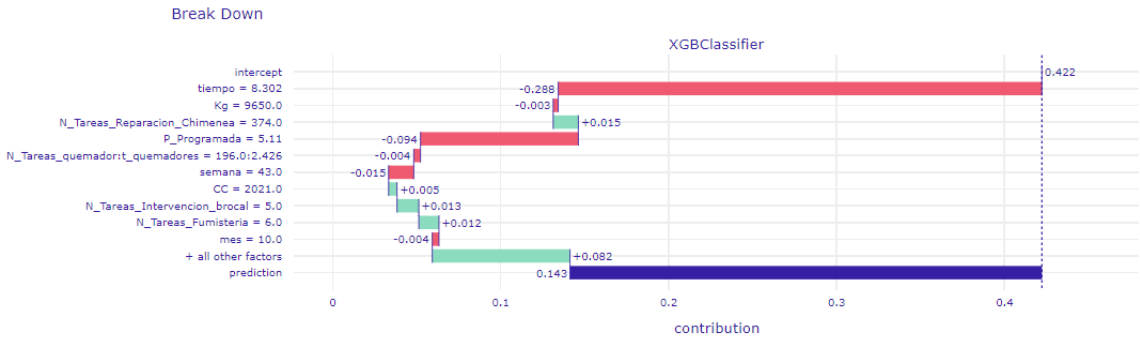

Figura 24: Aplicación de IBreakDown en un registro [400] del dataset testeo

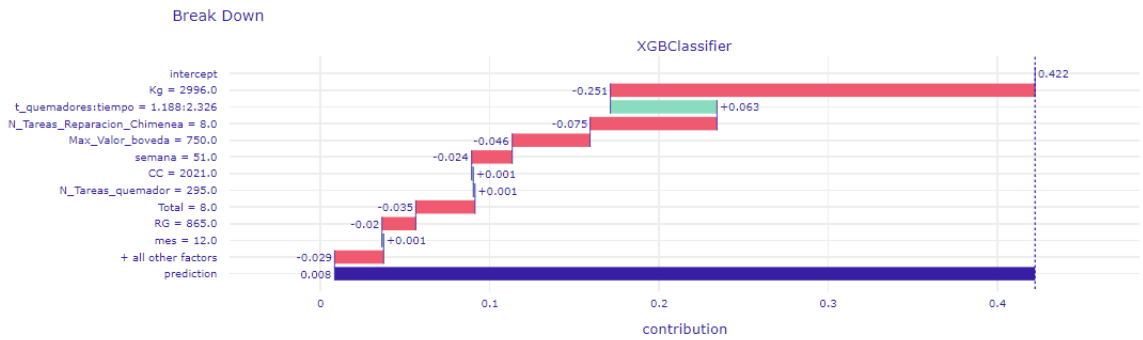

Figura 25: Aplicación de IBreakDown en un registro [500] del dataset testeo

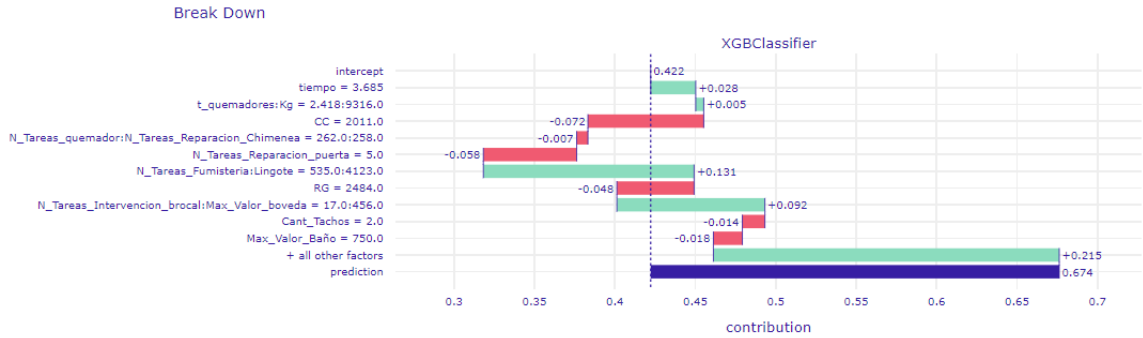

Figura 26: Aplicación de IBreakDown en un registro [550] del dataset testeo

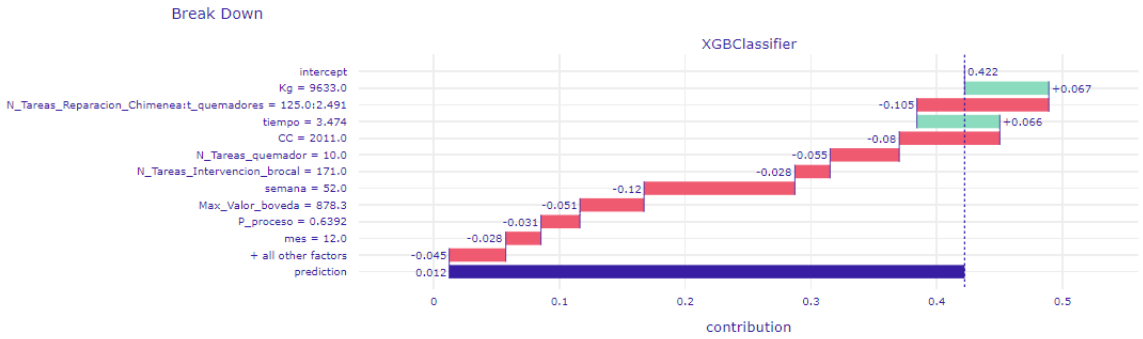

Figura 27: Aplicación de IBreakDown en un registro [600] del dataset testeo