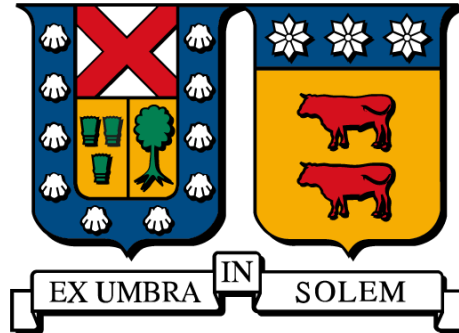


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Departamento de Ingeniería Metalúrgica y de Materiales



**CREACIÓN DE MODELOS DE APRENDIZAJE AUTOMÁTICO PARA LA
PREDICCIÓN DE POTENCIA EN MOLINO SAG A PARTIR DE ÍNDICES
DE CONMINUCIÓN DEL MINERAL Y OTRAS VARIABLES
OPERACIONALES**

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN MINAS

CARLOS IGNACIO OSES ÁVILA

PROFESORA GUÍA: FRANCISCA SAN MARTÍN ROBBIANO

OCTUBRE 2024

AGRADECIMIENTOS

Agradecer en primer lugar a Juan Carlos y Paola, mis padres, sin ellos, su amor incondicional y su apoyo nunca habría podido completar este proceso. También a Lusvenia, mi abuela, por estar siempre presente y por contagiarme la enorme fortaleza que ha demostrado este año frente a todo lo que ha tenido que vivir.

Agradecer también al resto de mi familia, en especial a mis hermanos Francisca y Gonzalo por darme mil vivencias que me han llevado a ser lo que soy hoy en día.

Por otra parte, agradecer a mi profesora guía Francisca San Martín por su compromiso y dedicación durante el desarrollo de mi trabajo de titulación.

De igual forma agradecer a las personas que formaron parte de mi vida durante esta larga etapa, pero por distintas circunstancias, los caminos se separaron. Agradecimientos especiales a Leslie Quezada por haber sido mi compañera a lo largo de años muy difíciles y por siempre brindarme su amor y la luz que muchas veces necesité para seguir avanzando.

Por último, agradecer a todos los verdaderos amigos que me acompañaron, por preocuparse, darme incontables alegrías y enseñarme sobre los altibajos de la vida. Todos ellos sabrán sus nombres al leer estas palabras.

RESUMEN

El presente trabajo tiene por objetivo el desarrollo de múltiples modelos de aprendizaje automático que sean capaces de predecir el comportamiento de la potencia eléctrica para los molinos SAG instalados en tres líneas de procesamiento distintas, a partir de información y datos obtenidos de la minera Los Pelambres. La motivación final de esta memoria es estudiar la viabilidad de una hipotética implementación de este tipo de herramientas en la industria, o en caso de que sea inviable, comprender mejor las falencias y requerimientos.

Para lograr el objetivo, se escogen como variables predictoras: *work index*, tonelaje diario procesado, P80 objetivo, porcentaje de sólidos y velocidad de giro del equipo. Se seleccionaron cuatro algoritmos para la creación de los modelos: Regresión lineal múltiple, *K- Nearest neighbor*, *Random forest* y *Gradient boosting*. Posteriormente se compararon los resultados mediante la magnitud de la raíz del error cuadrático medio (RMSE) y el índice de correlación entre la variable predicha y la real.

Los mejores resultados para las líneas SAG01 y SAG02 se obtuvieron mediante el algoritmo *Random forest*, el que arrojó un RMSE de 6,27% y un R^2 de 0,519 (para SAG01) y un RMSE de 5,09% y un R^2 de 0,598 (para SAG02). Por otra parte, para el caso del SAG03, *Gradient boosting (XGBoost)* obtuvo el mejor desempeño, entregando un RMSE de 4,92% y un R^2 de 0,465.

Finalmente, al analizar las variables utilizadas y los resultados de los modelos, se concluye que a pesar de que las predicciones fueron satisfactorias y con errores relativos cercanos al 6% del promedio de la variable objetivo, gran parte de su precisión es gracias al *input* “velocidad de giro del equipo”, variable que no es suficientemente anticipada durante la operación de los molinos. Por lo anterior, las estimaciones sin considerar esta variable no son confiables al punto de implementarlas en la supervisión o el control de esta etapa de la conminución de minerales. Sin embargo, se sugieren lineamientos para avanzar hacia el desarrollo de modelos más adecuados para este fin.

ABSTRACT

The aim of this study is developing multiple machine learning models capable of predicting the behavior of electrical power draw for SAG mills installed in three different processing lines, using information and data obtained from Los Pelambres mine. The ultimate motivation of this project is to investigate the feasibility of a hypothetical implementation of this kind of tools in the industry, or if unfeasible, to better understand the shortcomings and requirements.

To achieve the objective, the following predictor variables were chosen: work index, daily processed tonnage, target P80, percentage of solids and mill rotation speed. Four algorithms were selected for model creation: Multiple linear regression, K-Nearest neighbor, Random forest and Gradient boosting. Subsequently, the results were compared using the value of the root mean square error (RMSE) and the correlation index between the predicted and actual variable.

The best results for SAG lines 01 and 02 were obtained using the Random forest algorithm, which reached an RMSE of 6.27% and an R^2 of 0.519 (for SAG01), and an RMSE of 5.09% and R^2 of 0.598 (for SAG02). On the other hand, for SAG03 case, Gradient boosting (XGBoost) achieved the best performance, with an RMSE of 4.92% and an R^2 of 0.465.

Finally, upon analyzing the used variables and the results of the models, it is concluded that although the predictions were satisfactory with relative errors close to 6% of the target variable average, most of their accuracy is due to the “mill rotation speed” input, a variable that is not enough anticipated during mill operation. Therefore, estimations without considering this variable are not reliable enough to implement them in the supervision or control of this stage of mineral comminution. However, guidelines are suggested to advance towards the development of more suitable models for this purpose.

ÍNDICE

AGRADECIMIENTOS	2
RESUMEN	3
ABSTRACT	4
CAPITULO I: INTRODUCCIÓN	11
Objetivo general	11
Objetivos específicos	11
CAPÍTULO II: MARCO TEÓRICO	13
2.1 MOLIENDA SAG	13
2.1.1 Proceso	13
2.1.2 Parámetros y variables operacionales	15
2.1.2.1 P80	15
2.1.2.2 Indicadores de conminución	15
2.1.2.3 Capacidad de tratamiento	16
2.1.2.4 Porcentaje de sólidos	17
2.1.2.5 Velocidad de giro	17
2.1.2.6 Potencia	18
2.1.2.7 Agua alimentada	18
2.1.2.8 Consumo de bolas	18
2.1.3 Antecedentes Minera Los Pelambres (MLP)	18
2.1.4 Molienda MLP	19
2.2 ALGORITMOS PREDICTIVOS	20
2.2.1 Regresión lineal	21
2.2.2 K-Nearest neighbors (KNN)	22
2.2.3 Árbol de decisión	23
2.2.4 Random forest	24
2.2.5 Gradient boosting	25
2.3 EVALUACIÓN Y VALIDACIÓN DE MODELOS	26
2.3.1 Métricas de error en modelos	26
2.3.1.1 Mean Squared Error (MSE)	26
2.3.1.2 Root Mean Squared Error (RMSE)	26
2.3.1.3 Mean Absolute Error (MAE)	27
2.3.2 Métodos de validación de modelos	27

2.3.2.1	Validación cruzada (CV).....	27
2.3.2.2	K-Fold	27
2.3.2.3	Repeated K-Fold	27
2.3.2.4	GridSearch CV.....	27
2.4	ESTUDIOS PREVIOS RELACIONADOS.....	28
	CAPÍTULO III: METODOLOGÍA	29
	CAPÍTULO IV: RESULTADOS Y ANÁLISIS	33
4.1	ANÁLISIS EXPLORATORIO UNIVARIABLE Y LIMPIEZA DE DATOS	33
4.2	ANÁLISIS EXPLORATORIO MULTIVARIABLE.....	40
4.2.1	Indicadores de dureza de mineral	40
4.2.2	SAG 01.....	42
4.2.3	SAG 02.....	45
4.2.4	SAG 03.....	48
4.3	DESARROLLO Y EVALUACIÓN DE MODELOS.....	51
4.3.1	Método 1: Regresión lineal.....	51
4.3.1.1	SAG 01	52
4.3.1.2	SAG 02	53
4.3.1.3	SAG 03	53
4.3.2	Método 2: K-Nearest neighbors (KNN).....	54
4.3.2.1	SAG 01	55
4.3.2.2	SAG 02	55
4.3.2.3	SAG 03	56
4.3.3	Método 3: Random forest.....	57
4.3.3.1	SAG 01	58
4.3.3.2	SAG 02	58
4.3.3.3	SAG 03	59
4.3.4	Método 4: Gradient boosting	60
4.3.4.1	SAG 01	61
4.3.4.2	SAG 02	62
4.3.4.3	SAG 03	63
4.4	COMPARACIÓN DE MÉTODOS	64
	CAPÍTULO V: CONCLUSIONES	70
	REFERENCIAS	72

Índice de figuras

Figura 1: Componentes básicos de un molino SAG (Silva,2013).....	14
Figura 2: Fenómenos producidos al interior del molino SAG a distintas velocidades (N). a) N cercana a 0. b) N bajo el óptimo. c) N óptima. d) N crítica.	17
Figura 3: Ubicación y accesos de MLP.....	19
Figura 4: Esquema del proceso de molienda y clasificación de la planta Chacay.....	20
Figura 5: Representación gráfica de un conjunto de datos y su modelado mediante RL simple.	21
Figura 6: Representación gráfica del algoritmo KNN al clasificar un punto P.....	23
Figura 7: Representación gráfica de un modelo de árbol de decisión.....	23
Figura 8: Representación gráfica del algoritmo random forest, mostrando múltiples árboles de decisión.	24
Figura 9: Diagrama del proceso de bootstrapping (Amat, 2021).....	24
Figura 10: Representación gráfica del funcionamiento del algoritmo Gradient boosting.....	25
Figura 11: Diagrama resumen de la metodología aplicada en el estudio.	29
Figura 12: Ejemplo de matriz de correlación en un análisis multivariable.	32
Figura 13: Set de histogramas generados para las variables de la línea SAG 1 sin limpieza de datos.	34
Figura 14: Set de histogramas generados para las variables de la línea SAG 2 sin limpieza de datos.	35
Figura 15: Set de histogramas generados para las variables de la línea SAG 3 sin limpieza de datos.	35
Figura 16: Set de histogramas generados para las variables de la línea SAG 1 tras limpieza de datos.	38
Figura 17: Set de histogramas generados para las variables de la línea SAG 2 tras limpieza de datos.	39
Figura 18: Set de histogramas generados para las variables de la línea SAG 3 tras limpieza de datos.	39
Figura 19: Gráfico de dispersión para medidas de DWi y Axb.	41
Figura 20: Gráfico de dispersión para medidas de Axb y BWi.	41
Figura 21: Gráfico de dispersión para medidas de DWi y BWi.....	41
Figura 22: Mapa de calor con valores de correlación entre variables en el primer escenario.	42
Figura 23: Gráfico de dispersión entre Axb y la potencia del molino para el escenario 1.....	43
Figura 24: Gráfico de dispersión entre BWi y la potencia del molino para el escenario 1.	43
Figura 25: Gráfico de dispersión entre DWi y la potencia del molino para el escenario 1.	44
Figura 26: Gráficos de dispersión entre las distintas variables operacionales estudiadas y la potencia del molino para el escenario 1.	45
Figura 27: Mapa de calor con valores de correlación entre variables en el segundo escenario.	46
Figura 28: Gráfico de dispersión entre Axb y la potencia del molino para el escenario 2.....	46
Figura 29: Gráfico de dispersión entre BWi y la potencia del molino para el escenario 2.	47
Figura 30: Gráfico de dispersión entre DWi y la potencia del molino para el escenario 2.	47

Figura 31: Gráficos de dispersión entre las distintas variables operacionales estudiadas y la potencia del molino para el escenario 2.	48
Figura 32: Mapa de calor con valores de correlación entre variables en el tercer escenario.	49
Figura 33: Gráfico de dispersión entre A_{xb} y la potencia del molino para el escenario 3.	49
Figura 34: Gráfico de dispersión entre BW_i y la potencia del molino para el escenario 3.	50
Figura 35: Gráfico de dispersión entre DW_i y la potencia del molino para el escenario 3.	50
Figura 36: Gráficos de dispersión entre las distintas variables operacionales estudiadas y la potencia del molino para el escenario 3.	51
Figura 37: Distribución de valores reales vs predichos por RLM en escenario SAG 01, donde la línea punteada es la función identidad.	52
Figura 38: Distribución de valores reales vs predichos por RLM en escenario SAG 02, donde la línea punteada es la función identidad.	53
Figura 39: Distribución de valores reales vs predichos por RLM en escenario SAG 03, donde la línea punteada es la función identidad.	54
Figura 40: Distribución de valores reales vs predichos por KNN en escenario SAG 01, donde la línea punteada es la función identidad.	55
Figura 41: Distribución de valores reales vs predichos por KNN en escenario SAG 02, donde la línea punteada es la función identidad.	56
Figura 42: Distribución de valores reales vs predichos por KNN en escenario SAG 03, donde la línea punteada es la función identidad.	57
Figura 43: Distribución de valores reales vs predichos por RF en escenario SAG 01, donde la línea punteada es la función identidad.	58
Figura 44: Distribución de valores reales vs predichos por RF en escenario SAG 02, donde la línea punteada es la función identidad.	59
Figura 45: Distribución de valores reales vs predichos por RF en escenario SAG 03, donde la línea punteada es la función identidad.	60
Figura 46: Distribución de valores reales vs predichos por XGB en escenario SAG 01, donde la línea punteada es la función identidad.	61
Figura 47: Distribución de valores reales vs predichos por XGB en escenario SAG 02, donde la línea punteada es la función identidad.	62
Figura 48: Distribución de valores reales vs predichos por XGB en escenario SAG 03, donde la línea punteada es la función identidad.	63
Figura 49: Gráfico de potencia real vs predicha utilizando modelo RF para SAG 01 en un periodo de 6 meses.	66
Figura 50: Gráfico de potencia real vs predicha utilizando modelo RF para SAG 02 en un periodo de 6 meses.	66
Figura 51: Gráfico de potencia real vs predicha utilizando modelo XGB para SAG 03 en un periodo de 6 meses.	67
Figura 52: Gráfico de potencia real vs predicha utilizando modelo RF para SAG 01 sin considerar velocidad de giro como input.	68
Figura 53: Gráfico de potencia real vs predicha utilizando modelo RF para SAG 02 sin considerar velocidad de giro como input.	68
Figura 54: Gráfico de potencia real vs predicha utilizando modelo XGB para SAG 03 sin considerar velocidad de giro como input.	69

Figura 55: Primera sección de la Figura 49: Gráfico de potencia real vs predicha utilizando modelo RF para SAG 01 en un periodo de 6 meses.	73
Figura 56: Segunda sección de la Figura 49: Gráfico de potencia real vs predicha utilizando modelo RF para SAG 01 en un periodo de 6 meses.	73
Figura 57: Primera sección de la Figura 50: Gráfico de potencia real vs predicha utilizando modelo RF para SAG 02 en un periodo de 6 meses.	74
Figura 58: Segunda sección de la Figura 50: Gráfico de potencia real vs predicha utilizando modelo RF para SAG 02 en un periodo de 6 meses.	74
Figura 59: Primera sección de la Figura 51: Gráfico de potencia real vs predicha utilizando modelo XGB para SAG 03 en un periodo de 6 meses.....	75
Figura 60: Segunda sección de la Figura 51: Gráfico de potencia real vs predicha utilizando modelo XGB para SAG 03 en un periodo de 6 meses.....	75

Índice de tablas

Tabla 1: Sección de la base de datos utilizada para el estudio. Incluye variables de dureza, capacidad y granulometría.....	29
Tabla 2: Sección de la base de datos que muestra las variables operacionales disponibles del SAG 01.....	30
Tabla 3: Sección de la base de datos que muestra las variables operacionales disponibles del SAG 02.....	30
Tabla 4: Sección de la base de datos que muestra las variables operacionales disponibles del SAG 03.....	30
Tabla 5: Variables de la base de datos seleccionadas para el estudio y procesamiento.	31
Tabla 6: Estadística descriptiva de las variables seleccionadas para la base de datos de la línea SAG 1.....	33
Tabla 7: Estadística descriptiva de las variables seleccionadas para la base de datos de la línea SAG 2.....	33
Tabla 8: Estadística descriptiva de las variables seleccionadas para la base de datos de la línea SAG 3.....	34
Tabla 9: Cantidad inicial y final de observaciones para cada base de datos.....	36
Tabla 10: Percentiles de corte inferior y superior para las variables en la base de datos de la línea SAG01.	37
Tabla 11: Percentiles de corte inferior y superior para las variables en la base de datos de la línea SAG02.	37
Tabla 12: Percentiles de corte inferior y superior para las variables en la base de datos de la línea SAG03.	37
Tabla 13: Mejores hiperparámetros en modelo RF para SAG01. Estimados mediante GridSearch CV.	58
Tabla 14: Mejores hiperparámetros en modelo RF para SAG02. Estimados mediante GridSearch CV.	59
Tabla 15: Mejores hiperparámetros en modelo RF para SAG03. Estimados mediante GridSearch CV.	60

Tabla 16: Mejores hiperparámetros en modelo XGB para SAG01. Estimados mediante GridSearch CV.	61
Tabla 17: Mejores hiperparámetros en modelo XGB para SAG02. Estimados mediante GridSearch CV.	62
Tabla 18: Mejores hiperparámetros en modelo XGB para SAG03. Estimados mediante GridSearch CV.	63
Tabla 19: Valores de correlación entre potencia predicha y real para todos los escenarios.	64
Tabla 20: Valores de error porcentual en la estimación de la potencia, calculado a partir del RMSE en cada escenario.....	64

CAPITULO I: INTRODUCCIÓN

La conminución de minerales de cobre mediante el uso de molinos SAG se ha masificado enormemente en las últimas cuatro décadas debido a su capacidad de procesar grandes tonelajes. Sin embargo, esta misma característica junto al tipo de interacciones entre las partículas que circulan en su interior, provocan que este tipo de equipos sufra considerables pérdidas de energía, por lo que en estos términos no es considerado un proceso muy eficiente. Lo anterior se suma a que la fase de molienda de por sí, es la de más alto requerimiento energético dentro de las plantas de procesamiento. Por lo tanto, al poder disminuir esta variable, aunque sea en un punto porcentual, se puede generar una oportunidad de disminuir también los costos operativos totales de la industria. Una buena oportunidad para lograr esto, es el uso de las recientes tecnologías de procesamiento de grandes cantidades de datos junto con algoritmos de aprendizaje automático o *machine learning*.

Bajo la premisa anterior, este trabajo busca realizar un primer acercamiento al estudio de distintos algoritmos predictivos y a la aplicación de estos en la estimación de variables operativas de los equipos SAG, en este caso específico, la potencia de estos. Lo anterior se realiza a partir de *inputs* o variables independientes, poniendo especial foco en el efecto que tienen la dureza del mineral alimentado, reflejado en los valores de sus índices de conminución.

Por último, se busca comparar los resultados de los distintos modelos y obtener conclusiones que permitan acercarse a la idea de una posible implementación de este tipo de tecnologías a los sistemas de control utilizados en la industria.

Objetivo general

Desarrollar y estudiar la viabilidad del uso de distintos modelos de aprendizaje automático para la predicción de la potencia eléctrica en molienda SAG a partir de índices de conminución y otras variables operacionales.

Objetivos específicos

- Analizar el estado de arte relacionado al impacto de las características del mineral en las variables operacionales de la etapa de molienda.
- Crear y validar varios modelos para tres escenarios de equipos SAG, utilizando cuatro algoritmos distintos e *inputs* como: *work index*, tonelaje procesado, P80 objetivo, porcentaje de sólidos y velocidad de giro del equipo.

- Analizar y generar conclusiones a partir de los resultados entregados por los distintos modelos.

CAPÍTULO II: MARCO TEÓRICO

2.1 MOLIENDA SAG

2.1.1 Proceso

Dentro del área de procesamiento de minerales en la gran minería, los circuitos de molienda que incluyen molinos SAG son los más aceptados y adoptados en las últimas décadas. Como menciona Silva (2013) la molienda convencional tiene asociada un alto costo operacional, caracterizado por el consumo de acero de los medios de molienda y de los revestimientos del equipo. Las tasas de desgaste incrementales y los costos operacionales asociados, llevaron a la industria minera a buscar nuevas metodologías que hicieran frente a este aspecto. En respuesta a ello surgió la idea de reducir e incluso evitar el uso de los medios de molienda de acero al interior de los molinos, llegando con ello al desarrollo de grandes equipos, similares a los convencionales, que pudiesen ocupar las mismas colpas de mineral como remplazo a las bolas de acero originales. Este es el principio fundamental en el que se basa el diseño de estos equipos, que nacen como una variante de los molinos autógenos (AG). En la Figura 1 se muestra un esquema del equipo SAG y sus principales componentes a modo de referencia.

Lo que diferencia al molino SAG del AG es la adición de un medio de molienda, en este caso una carga de bolas de acero, a pesar de que esto adiciona costos, el SAG posee significativas ventajas por sobre sus antecesores, entre las que destacan:

- Mayor eficiencia energética del proceso ya que las bolas de acero permiten alcanzar una menor granulometría del mineral en menos tiempo, es decir, el material suele tener un menor tiempo de residencia dentro del equipo.
- Menor desgaste de revestimientos debido a que una parte de los impactos de las colpas de mineral es absorbida por los medios de molienda.

En un circuito típico, el molino SAG procesa el material proveniente de chancado primario y lo prepara, reduciendo la granulometría hasta lo que requiere la etapa posterior de flotación. Normalmente, el producto del equipo se somete a tamizado mediante harneros de gran tamaño para separar las partículas con tamaño adecuado del mineral sobredimensionado. Las partículas de tamaño adecuado se dirigen hacia molinos de menor capacidad, normalmente molinos de bolas para una última fase de molienda. El mineral sobredimensionado se suele recircular hacia chancadores de baja capacidad y luego se reintroduce junto a la alimentación del molino SAG. Este proceso optimiza la cantidad de mineral alimentado y reduce al mínimo el consumo de energía eléctrica.

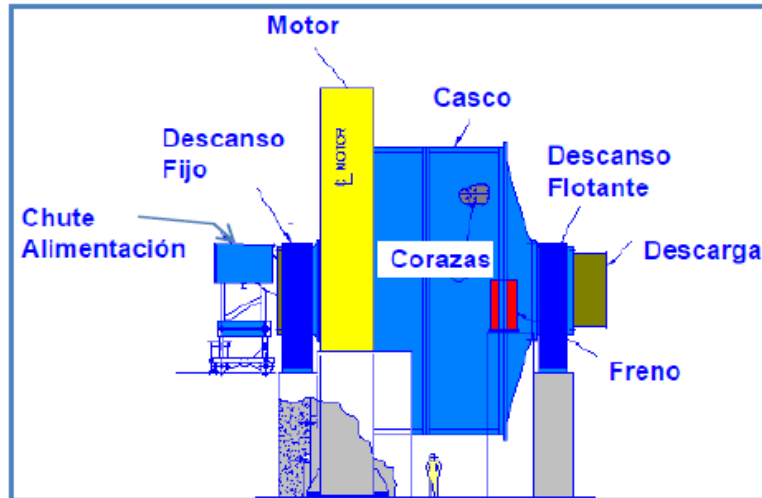


Figura 1: Componentes básicos de un molino SAG (Silva, 2013).

La tendencia a la baja en las leyes de la minería del cobre y al aumento de la dureza de los minerales procesados (en gran parte debido a la profundización de los yacimientos) han llevado a una notable masificación del uso de estos equipos dentro de la industria de este metal, esto, debido a su capacidad de procesar grandes volúmenes de material y su adaptabilidad a variaciones de dureza en la alimentación. Adicionalmente, se ha observado una considerable baja en el costo de capital a invertir al compararlo con un circuito convencional que incluye las etapas de chancado secundarios y/o terciario para un mismo rango de capacidades de procesamiento.

Por otra parte, el gran tamaño de las colpas alimentadas a este proceso conlleva a que gran parte de la energía suministrada sea disipada en interacciones entre estas mismas o en movimientos giratorios de cada una, por lo que el consumo energético de los molinos SAG es bastante elevado (Silva, 2013). Además, considerando que la etapa de molienda es la de mayor requerimiento energético específico dentro de la planta de procesamiento, la fracción utilizada por los equipos en cuestión puede llegar a ser muy significativa respecto al total requerido para llegar al producto final.

Es importante también destacar un fenómeno que se produce en estos circuitos, como se menciona en el Manual general de minería y metalurgia (Portal minero, 2006), en los casos más comunes donde se ha implementado la tecnología de molienda AG y SAG, se ha observado una condición en la que los minerales alimentados directamente desde un chancador primario giratorio generan un tamaño crítico de colpas que no se someten a molienda y simplemente se redondean dentro del molino, siendo llamados "*pebbles*". Este problema se ha abordado mediante la instalación de rejillas en la salida del molino con orificios apropiados para eliminar estos elementos y triturarlos de manera separada en un

chancador de conos dedicado exclusivamente a estos *pebbles*. La granulometría de estas partículas se encuentra usualmente entre 1” y 3”.

2.1.2 Parámetros y variables operacionales

Los principales parámetros operacionales para considerar en el funcionamiento de los molinos semi autógenos son:

2.1.2.1 P80

Es una medida estadística muy usada en el procesamiento de minerales y entrega información granulométrica de la muestra, específicamente, es el tamaño de partícula bajo el cual está el 80% (en masa) de la muestra. Para procesos de molienda se suelen utilizar los micrones (μm) como unidad de medida.

El P80 indica un tamaño característico para una muestra de descarga de una etapa del proceso. Por otra parte, el llamado F80 representa lo mismo, pero en una muestra de la alimentación. Determinar este parámetro puede proporcionar información sobre la eficiencia de alguna etapa dentro del circuito de conminución y su razón de reducción.

2.1.2.2 Indicadores de conminución

Esta variable viene dada por la mineralogía de los minerales que son procesados, que a su vez depende del tipo de yacimiento y de los metales de interés que se buscan extraer. Múltiples estudios indican que es un factor de gran importancia en la variabilidad de la potencia de los molinos SAG y su consumo energético (Ver sección 2.4). Los principales parámetros que se usan en la industria, y en especial para los procesos de molienda, son: BWi, DWi, Axb, los cuales son detallados a continuación:

- **BWi:** El *Bond work index* es un indicador de la resistencia de un mineral a ser triturado mediante chancado o molienda. Fue desarrollado por Fred Bond y se determina mediante el *test* de Bond, el cual se realiza moliendo una muestra en un molino de laboratorio hasta que el P80 alcance una medida de tamiz de 100 μm para posteriormente registrar la energía utilizada y se expresa en kWh/t. Lo anterior tiene su origen en el modelo de Bond, representado por la siguiente ecuación:

$$W = W_i * \left(\frac{10}{\sqrt{P_{80}}} - \frac{10}{\sqrt{F_{80}}} \right)$$

De esta forma es posible calcular la energía específica requerida por el molino (W en kWh/t) para triturar el mineral desde un tamaño conocido (P_{80}) hasta uno requerido (F_{80}) (Both y Dimitrakopoulos, 2021). Por lo tanto, el *work index* de Bond (W_i o BW_i) se define numéricamente como la energía que se requiere por tonelada de

mineral para reducir la muestra desde un tamaño teóricamente infinito en la alimentación hasta un P80 de 100 μm en la descarga.

El BWi se utiliza para comparar distintos materiales procesados, por lo que puede ser considerado un indicador de dureza relativa. Entre mayor es este indicador, mayor energía se requiere para reducir de tamaño la muestra. Adicionalmente, tiene aplicaciones como el dimensionamiento de equipos para planta y su uso como parámetro para una óptima operación de estos.

- **DWi:** El *Drop weight index* es un indicador de la resistencia al impacto de los minerales. Este es determinado mediante el *test* de caída SMC en una máquina de ensayo estandarizada, donde se libera una masa conocida sobre una muestra de mineral con el fin de disminuir su granulometría solo por efecto de su peso cayendo desde una altura conocida. Luego, con la energía de impacto registrada y la nueva distribución granulométrica de la muestra, se calcula el DWi, medido en kWh/m^3 (Verret et al., 2011). Numéricamente se define como la energía total de impacto suministrada a una muestra por cada unidad de volumen de esta (m^3).

Este índice puede ser útil al dimensionar circuitos de molienda y chancado, pero también es ampliamente utilizado para medir la variabilidad dentro de las distintas zonas de los yacimientos.

- **Axb:** Este parámetro de conminución se obtiene de un ensayo que consiste en tres secciones donde se miden distintas características de una muestra mineral. Primero una prueba de resistencia al impacto basada en el Drop weight tester que determina el DWi. Seguida de una evaluación de rotura de las partículas por abrasión y, por último, una medición de densidad de la muestra. De esta forma el ensayo genera tres parámetros: A, b y t_a (Olmos, 2018), con los cuales se calcula el parámetro Axb, que es adimensional e inversamente proporcional a la resistencia a la molienda del mineral. Es utilizado principalmente en dimensionamiento y optimización de procesos de conminución.

2.1.2.3 Capacidad de tratamiento

Se define como el flujo másico de mineral (t/h) que circula por el molino SAG en un tiempo dado. Este está limitado por la capacidad nominal del equipo, que a su vez viene dado principalmente por las dimensiones de este. Adicionalmente, este parámetro se puede aproximar al obtener el tonelaje total procesado (t) en un periodo conocido.

2.1.2.4 Porcentaje de sólidos

Es el porcentaje de masa de mineral respecto al total en masa de pulpa que es alimentada al molino. Al igual que la dureza del mineral, este parámetro ha mostrado una fuerte incidencia en el consumo energético de los equipos, ya que está directamente relacionado con la viscosidad de la pulpa y, por lo tanto, con las fuerzas ejercidas entre partículas. Considerando m_s como la masa de sólidos en la pulpa y m_a como la masa de agua presente en la misma, esta variable se calcula como:

$$\frac{m_s}{m_s + m_a} * 100\%$$

2.1.2.5 Velocidad de giro

Es la velocidad rotacional del molino SAG, normalmente se expresa en revoluciones por minuto. Este parámetro influye directamente en la eficiencia del proceso, ya que, dependiendo de su cercanía a la velocidad crítica del equipo, puede incluso evitar que se lleve a cabo el proceso de conminución.

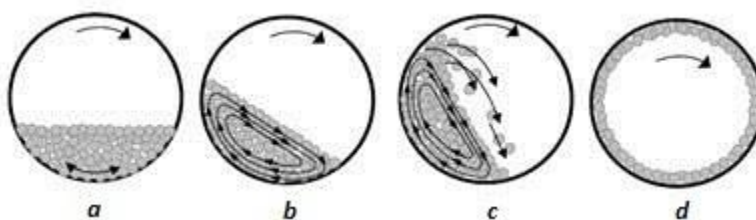


Figura 2: Fenómenos producidos al interior del molino SAG a distintas velocidades (N). a) N cercana a 0. b) N bajo el óptimo. c) N óptima. d) N crítica.

La velocidad crítica depende únicamente de las dimensiones del molino y se define como la velocidad a la que los medios de molienda y el mineral alimentado, es decir, la carga, no se desprende del manto debido a la fuerza centrífuga. Este parámetro se calcula como:

$$N_{crit} = \frac{76,6}{\sqrt{D}}$$

Donde:

N_{crit} = Velocidad crítica en rpm.

D = Diámetro del molino SAG en pies.

La Figura 2 muestra un esquema del comportamiento del molino y su carga para distintos valores de este parámetro (N).

Otro parámetro que se utiliza en lugar de la velocidad directa del molino en rpm, pero que cumple la misma función, es η que se calcula como:

$$\eta = \frac{N}{N_{crit}}$$

Donde:

N = Velocidad a la que opera el molino en rpm.

N_{crit} = Velocidad crítica en rpm.

2.1.2.6 Potencia

La potencia es la cantidad de energía que se requiere para hacer funcionar un molino y triturar una determinada masa mineral hasta un tamaño específico, suele ser medida en kW o HP (caballos de fuerza). Es un parámetro de gran importancia en el dimensionamiento de los equipos, por lo que usualmente se estima a partir de la ecuación de Bond (Ver sección 2.1.2.2) utilizando indicadores como el BWi.

Este parámetro depende principalmente del tipo de equipo en uso y del resto de las variables operacionales (características del mineral procesado, velocidad de giro, tipo y desgaste de revestimientos, niveles de llenado y carga, entre otros).

2.1.2.7 Agua alimentada

Es el volumen de agua que ingresa al equipo de molienda junto al mineral procesado con el fin de generar una carga de tipo pulpa. Se suele medir en metros cúbicos (m^3) y está directamente relacionado con el nivel de llenado del molino y con el porcentaje de sólidos de la carga.

2.1.2.8 Consumo de bolas

El consumo de bolas es la cantidad de medios de molienda de acero que se adicionan a la carga del molino dentro de un periodo de tiempo definido, suele medirse en toneladas diarias (t/día). Las bolas de acero deben ser constantemente repuestas debido al desgaste y fractura que pueden presentar durante la operación. Este parámetro también es de gran importancia al evaluar los costos operacionales de la etapa de molienda.

2.1.3 Antecedentes Minera Los Pelambres (MLP)

Minera Los Pelambres (MLP) es una mina de cobre a cielo abierto que se ubica cercana a la ciudad de Salamanca, dentro de la IV región (Ver Figura 3) y pertenece en su mayoría al grupo Antofagasta Minerals (60%). En 2022 tuvo una producción de 275.000 toneladas del

metal rojo además de 7.200 toneladas de molibdeno y 43.100 onzas de oro como subproductos. Según la información de consejo minero, en el año ya mencionado se promedió una dotación de 7.508 trabajadores, entre internos y contratistas.

En cuanto al yacimiento explotado este es “de tipo pórfido cuprífero adakítico, cuya mineralización corresponde principalmente a sulfuros de cobre, con porcentajes bajos de molibdeno y valores subordinados de oro y plata” (Saez, 2009). Según el mismo autor, la geología del área indica cuerpos de diorita cuarcífera, pórfido docético y pórfido andesítico, con calidades geotécnicas entre buenas y regulares según la clasificación de Laubscher (1990).

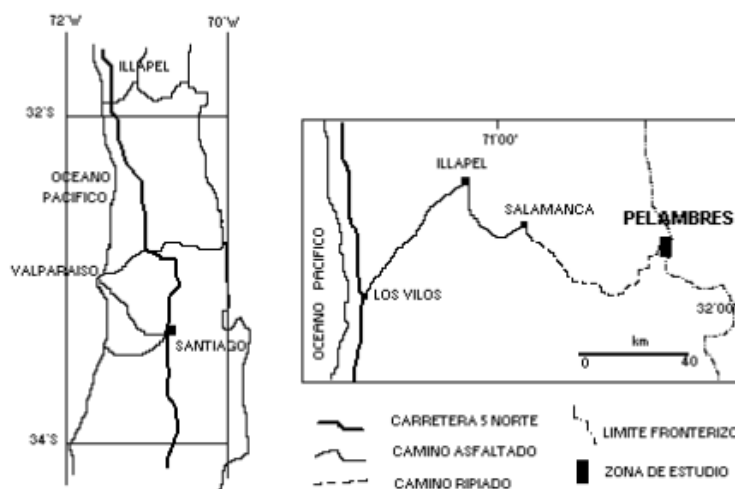


Figura 3: Ubicación y accesos de MLP.

2.1.4 Molienda MLP

El mineral que se extrae en el rajo es enviado a la planta de procesamiento Chacay (Ver Figura 4), donde primero es almacenado en un *stockpile* con capacidad aproximada de 480.000 toneladas para posteriormente ser alimentado a la etapa de molienda. Esta etapa se compone por tres líneas en paralelo que abastecen en conjunto el posterior proceso de flotación. Cada línea comienza con un molino SAG de 36 x 17 pies y de 15.500 kW de potencia, los que pueden llegar a procesar 3.000 toneladas por hora (Morell, 2014).

El circuito posee adicionalmente una planta de pebbles y una planta de gravilla, a utilizarse según la reducción de tamaño requerida para el material que no se puede conminuir dentro de los equipos semiautógenos.

Luego se procede a una clasificación del mineral mediante baterías de hidrociclones dispuestas en el flujo de salida de cada molino principal, el *underflow* es dirigido a la etapa de molienda secundaria, mientras que el *overflow* pasa directamente a flotación.

La molienda secundaria se compone principalmente por seis molinos de bolas (dos por cada línea de procesamiento) estos trabajan en circuito cerrado con los hidrociclones mencionados anteriormente (Pasten, 2018). En esta parte se cuenta con molinos de 21 x 33 pies y potencia de 7.200 kW y otro de 26 x 40 pies con potencia de 14.000 kW. El objetivo final es lograr un producto con tamaño de partícula cercano a las 200 micras.

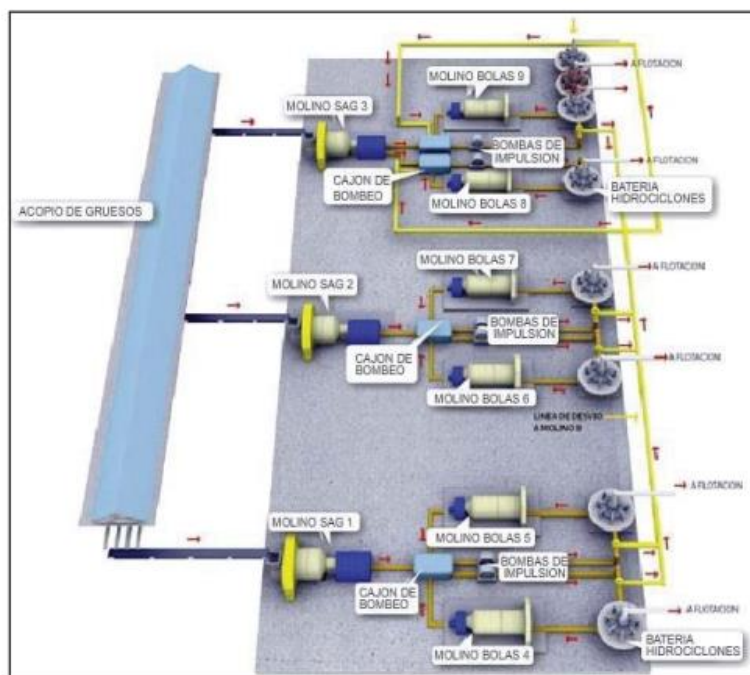


Figura 4: Esquema del proceso de molienda y clasificación de la planta Chacay.

2.2 ALGORITMOS PREDICTIVOS

Para la presente memoria de título es necesario conocer y definir algunos conceptos acerca de los algoritmos que serán utilizados durante el desarrollo experimental y posteriormente comparados para definir el que mejor se ajuste a los objetivos planteados.

En la aplicación de aprendizaje automático existen dos tipos de algoritmos, por un lado, los de regresión, que busca predecir valores continuos a partir de una serie de *inputs*, mientras que por el otro están los de clasificación que buscan asignar una clase o categoría a un

conjunto de datos. Para fines prácticos de este estudio, se profundizará exclusivamente en algunos de los primeros (algoritmos de regresión), debido a que se espera conocer el comportamiento de variables operacionales continuas de los molinos SAG (Potencia, consumo de bolas, etc.).

2.2.1 Regresión lineal

La regresión lineal es un algoritmo que busca modelar la relación que existe entre una o más variables independientes (regresores) y la variable dependiente (respuesta) realizando el ajuste de una ecuación lineal (Rojo y Abuín, 2007). Asignando $x_1, x_2, \dots, x_n$ a los valores de los regresores e y a la variable de respuesta se puede describir un modelo de regresión lineal mediante la siguiente ecuación:

$$y = b_0 + b_1x_1 + b_2x_2 + \dots + b_nx_n$$

Donde $b_1, b_2, \dots, b_n$ son conocidos como coeficientes de regresión y asignan una ponderación o “importancia” a cada variable explicativa. Estos valores se eligen de tal forma que se minimice la varianza residual dada por:

$$\sum (y_j - \hat{y}_j)^2$$

Donde y_j es el valor real de la variable dependiente e $\hat{y}_j$ es el valor predicho por el modelo. La sumatoria se realiza para todas las observaciones estudiadas.

Al algoritmo se le asigna el nombre de regresión lineal simple (Ver Figura 5) o múltiple dependiendo de si se considera una variable explicativa o múltiples variables, respectivamente. Para el primer caso el modelo queda reducido a la siguiente expresión:

$$y = b_0 + b_1x$$

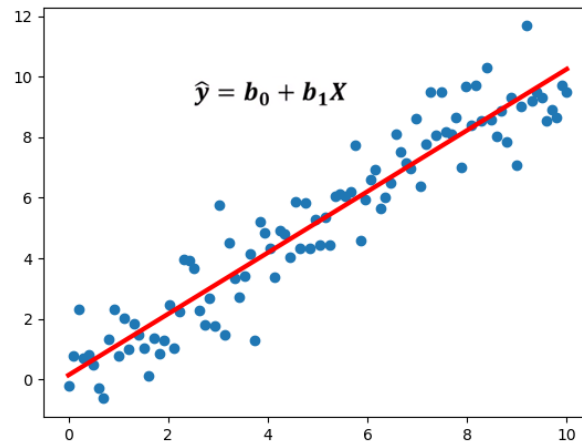


Figura 5: Representación gráfica de un conjunto de datos y su modelado mediante RL simple.

Adicionalmente, existen variaciones de modelos para la regresión lineal múltiple, que se caracterizan por principalmente reducir el sobreajuste (u *overfitting*) a un conjunto de datos y reducir el efecto de uno o más regresores con alta correlación o dependencia entre ellos. Dos de estas variaciones son:

- Modelo Lasso (*Least Absolute Shrinkage and Selection Operator*), que incorpora una regulación que penaliza la suma total de los valores absolutos de los coeficientes de regresión, por lo tanto, tiende a forzar algunos de estos a cero, anulando la influencia de la variable que lo acompaña, a esta penalización se le conoce como L1 y puede llevar a mejorar la generalización del modelo.
- Modelo Ridge, de forma similar al modelo Lasso, aplica una regulación, pero esta penaliza la suma total de los cuadrados de los coeficientes de regresión (penalización L2), por lo tanto, tiende a reducir el valor de estos, pero sin llegar a anularlos, considerando todas las variables explicativas que sean entregadas como *inputs*.

2.2.2 K-Nearest neighbors (KNN)

El algoritmo de *K-nearest neighbors* se basa en predecir el valor de la variable objetivo en una observación identificando las K observaciones del conjunto de entrenamiento que sean más semejantes a ella. Esto lo realiza calculando una distancia (normalmente euclidiana) a partir de los predictores o regresores (Ver Figura 6). Una vez seleccionadas las K observaciones, se predice el nuevo valor utilizando el promedio de estas.

Para este método es importante normalizar los valores de los regresores previamente ya que de esta forma cada variable independiente contribuirá de la misma manera en el cálculo de la distancia entre observaciones. También es importante una correcta elección del valor de K, un valor muy pequeño tiende a producir *overfitting* en el modelo, mientras que uno muy alto tiende a generalizarlo demasiado.

K Nearest Neighbors

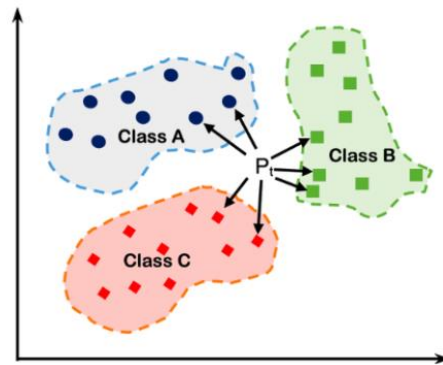


Figura 6: Representación gráfica del algoritmo KNN al clasificar un punto P_i .

2.2.3 Árbol de decisión

El árbol de decisión es un modelo predictivo que se forma por reglas binarias (Verdadero o Falso) y que tiene estructura de árbol, de esta manera las observaciones son repartidas en función de sus variables independientes o regresores. Las decisiones o condiciones son representadas por nodos, mientras que las ramas indican salidas o resultados (Ver Figura 7). Existen nodos raíz, que muestran la condición inicial del modelo, nodos internos, que son preguntas sobre atributos del conjunto de datos y lo dividen en subconjuntos de menor tamaño, y, por último, los nodos hoja o terminales, que indican la etiqueta asignada por el modelo a una determinada observación, en algoritmos de regresión entregan un valor numérico.

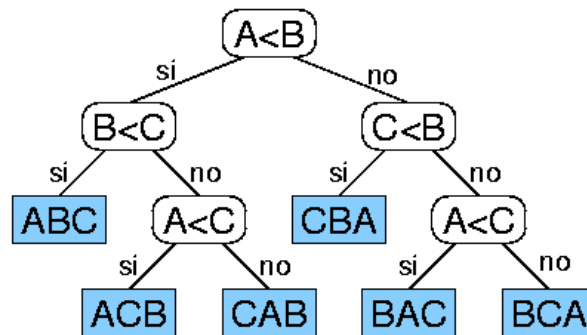


Figura 7: Representación gráfica de un modelo de árbol de decisión.

Este tipo de modelos se pueden representar gráficamente, aun con una cantidad de regresores muy grande y también son el elemento básico de modelos predictivos con mayor complejidad como *Random forest* y *Gradient boosting*.

2.2.4 Random forest

Este tipo de algoritmo se basa principalmente en la creación de múltiples árboles de decisión (Ver Figura 8 y sección 2.2.3), entrenados con distintas muestras aleatorias obtenidas de la muestra de entrenamiento original. Random forest es considerado un algoritmo de ensamble, es decir, combina las predicciones de varios modelos más básicos para reducir el efecto de *overfitting*. Como valor final de la variable objetivo se toma la media de todas las predicciones realizadas en los distintos arboles de decisión.

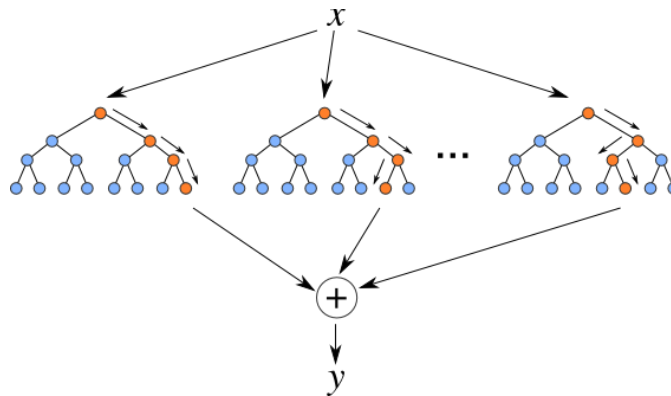


Figura 8: Representación gráfica del algoritmo random forest, mostrando múltiples árboles de decisión.

Las muestras aleatorias que se utilizan para el entrenamiento de los múltiples arboles de decisión son obtenidas mediante el método de *bootstrapping*, el que se basa en la generación de pseudo-muestras a partir de muestreo con reemplazo en la muestra original. Esto considerando que, si la muestra original de la población es representativa de esta, la distribución estadística que se obtiene mediante este método se aproximará a la distribución que se podría obtener en el caso de poder generar múltiples muestras a partir de la población completa. El proceso de este método queda explicado conceptualmente en la Figura 9.

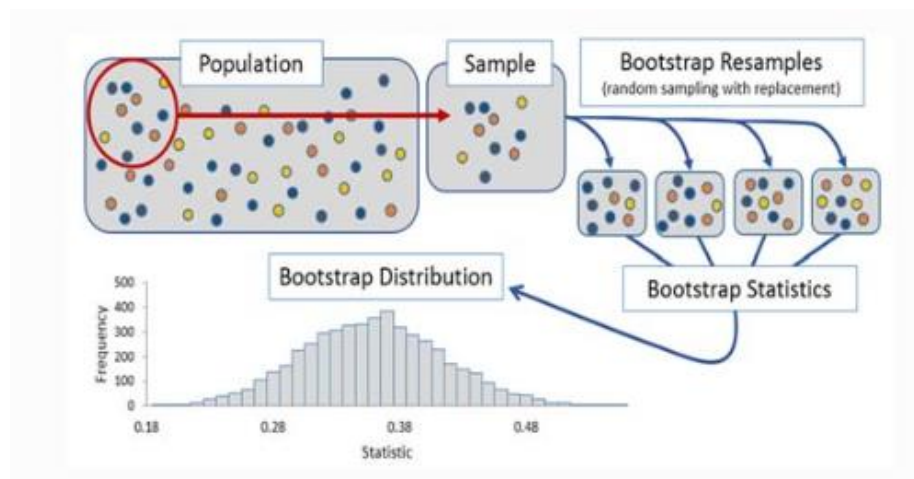


Figura 9: Diagrama del proceso de bootstrapping (Amat, 2021).

Los principales hiperparámetros que ajustan este algoritmo y sus definiciones son los siguientes:

- *Max depth*: Es la profundidad máxima de cada árbol de decisión generado. Controla la cantidad de ramificaciones que se puede hacer en cada uno.
- *Min samples leaf*: Es el número mínimo de observaciones que debe tener un nodo terminal.
- *Min samples split*: Es el número mínimo de observaciones que debe tener un nodo interno para dividirse.
- *N estimators*: Es el número de árboles de decisión que generará el modelo.

2.2.5 Gradient boosting

La técnica de aprendizaje automático *Gradient boosting* se basa en la idea de desarrollar un modelo final a partir de múltiples modelos de predicción de menor complejidad o predictores débiles, que al igual que el algoritmo *Random Forest*, generalmente utiliza árboles de decisión para este propósito (Ver sección 2.2.3). El modelo utiliza estos predictores de forma iterativa buscando reducir los residuos del predictor anterior, es decir, minimizar la diferencia entre los valores reales y los predichos (Ver Figura 10).

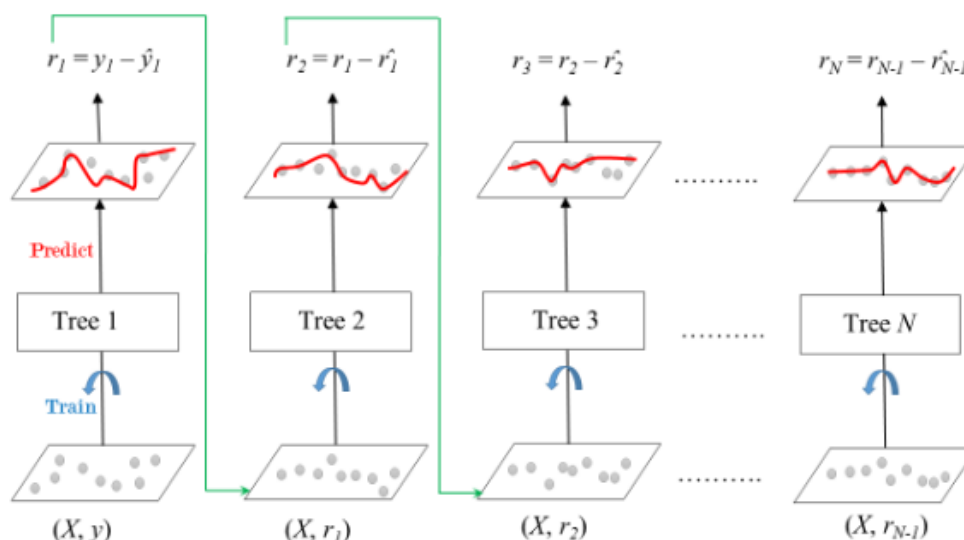


Figura 10: Representación gráfica del funcionamiento del algoritmo Gradient boosting.

Existen diversos algoritmos de *Gradient boosting*, entre los más populares se encuentran: *XGBoost*, *LightGBM* y *Gradient Boosting Machine*. Este último es la versión base del algoritmo, mientras que los dos primeros solo difieren en la eficiencia al utilizar los recursos computacionales en su entrenamiento-prueba y en la velocidad de entrenamiento de los modelos.

Los principales hiperparámetros que ajustan este algoritmo y sus definiciones son los siguientes:

- *Learning rate*: Es la tasa de aprendizaje que trabaja actualizando los pesos de cada árbol en cada iteración. Entre menor sea, más tiempo requiere el entrenamiento del modelo.
- *Max depth*: Es la profundidad máxima de cada árbol de decisión generado. Controla la cantidad de ramificaciones que se puede hacer en cada uno.
- *N estimators*: Es el número de árboles de decisión que generará el modelo.
- *Subsample*: Es la proporción de muestras que se utiliza para entrenar cada árbol.
- *Colsample bytree*: Es la proporción de atributos que se utiliza para entrenar cada árbol.

2.3 EVALUACIÓN Y VALIDACIÓN DE MODELOS

2.3.1 Métricas de error en modelos

Con el fin de medir el rendimiento de los distintos algoritmos y adicionalmente poder compararlos entre ellos es necesario establecer métricas de error para cada uno. Las métricas más utilizadas para problemas de regresión de datos son las definidas a continuación:

2.3.1.1 Mean Squared Error (MSE)

Esta métrica se calcula como la media de los errores elevados al cuadrado, donde el error es la diferencia del valor real con el valor predicho por el modelo. Se suele utilizar como un indicativo de cómo funciona el modelo en general, pero al estar en unidades cuadradas no tiene sentido físico en cuanto a las variables estudiadas. El MSE se puede calcular mediante la siguiente fórmula:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \tilde{y}_i)^2$$

Donde n es el número de observaciones, y_i el valor real de la variable e $\tilde{y}_i$ el valor estimado por el modelo.

2.3.1.2 Root Mean Squared Error (RMSE)

Para una mejor interpretación del MSE se puede recurrir al *Root Mean Squared Error* (RMSE), que se define como la raíz del MSE y sus unidades son las mismas que las de la variable objetivo. Se puede calcular como:

$$RMSE = \sqrt{MSE}$$

2.3.1.3 Mean Absolute Error (MAE)

Se calcula como la media aritmética de los valores absolutos de los errores (diferencia del valor real con el valor predicho). A diferencia del MSE, esta medición de error le da una ventaja a los modelos que tienen desviaciones más similares u homogéneas en todas las observaciones, ya que el primero al elevar al cuadrado los errores, penaliza más las mayores distancias entre la estimación de la variable objetivo y su valor etiquetado. El MAE se representa por la siguiente ecuación:

$$MSE = \frac{1}{n} \sum_{i=1}^n |y_i - \tilde{y}_i|$$

Donde n es el número de observaciones, y_i el valor real de la variable e $\tilde{y}_i$ el valor estimado por el modelo.

2.3.2 Métodos de validación de modelos

2.3.2.1 Validación cruzada (CV)

La validación cruzada es un método ampliamente utilizado en la evaluación y validación de modelos *machine learning*.

2.3.2.2 K-Fold

El método de validación cruzada *K-Fold* consiste en la división del conjunto de datos en K subconjuntos con igual cantidad de observaciones. Luego se entrena el modelo K veces, utilizando en cada una un subconjunto (*fold*) diferente como set de prueba y el resto como set de entrenamiento. Al terminar con las K iteraciones, promedia las métricas de evaluación (como puede ser MSE) para obtener una estimación del rendimiento general del modelo.

2.3.2.3 Repeated K-Fold

Repeated K-Fold trabaja de forma similar al *K-Fold* convencional, con la única diferencia de realizar múltiples iteraciones con diferentes particiones aleatorias de los datos. Lo anterior ayuda a obtener una estimación más fiable del rendimiento de los distintos modelos.

2.3.2.4 GridSearch CV

GridSearch CV o Validación cruzada con búsqueda en rejilla es una técnica utilizada para encontrar los hiperparámetros que mejor se adaptan a cada modelo. Esto lo realiza mediante múltiples iteraciones, cada una combinando distintos valores de cada parámetro, los cuales se especifican en una matriz. Para cada combinación se evalúa el rendimiento del algoritmo

mediante validación cruzada, utilizando, por ejemplo, *K-Fold*, para finalmente, entregar los mejores valores para el entrenamiento del modelo desarrollado.

Esta técnica se recomienda principalmente para algoritmos que posean una gran cantidad de hiperparámetros como, por ejemplo, *Random Forest* y *Gradient Boosting* (Ver secciones 2.2.4 y 2.2.5, respectivamente).

2.4 ESTUDIOS PREVIOS RELACIONADOS

La gran importancia económica de la utilización de la energía en los equipos de molienda ha impulsado diversos estudios que abarcan la relación que existe entre parámetros como la resistencia de los minerales procesados y la potencia requerida por los molinos.

En Wambeke et al. (2018) se realiza un modelo que permite actualizar la información de los índices BWi ya presentes en el modelo de bloques a partir de los datos de funcionamiento de los molinos de bolas instalados en la planta, más específicamente, realiza una predicción de cómo se deberían comportar los molinos en variables como el tonelaje procesado y la potencia consumida según los datos geometalúrgicos ya existentes. Para luego realizar una conciliación entre esta estimación y los datos reales, ajustando de esta manera los índices de resistencia a la conminución (BWi) de las unidades geológicas presentes en el modelo del yacimiento. Finalmente, se proyectan mejoras en la producción de hasta un 26%.

Cortés (2018), propone la realización de modelos estadísticos que sean capaces de predecir el rendimiento de la planta a partir del índice Axb e índices granulométricos (F80 y %1,25) por unidad geometalúrgica. Construyendo el modelo a partir de los datos a lo largo de dos años de operación. Finalmente, los modelos se utilizan para cuantificar el efecto económico que tuvo una etapa de pre-chancado en un periodo aproximado de tres meses. Sin embargo, se menciona que no se logra capturar de manera satisfactoria la variabilidad del rendimiento de la planta, por lo que se recomienda utilizar datos más actuales.

CAPÍTULO III: METODOLOGÍA

Para la presente memoria se propone la metodología detallada a continuación y que se ve representada en la Figura 11.

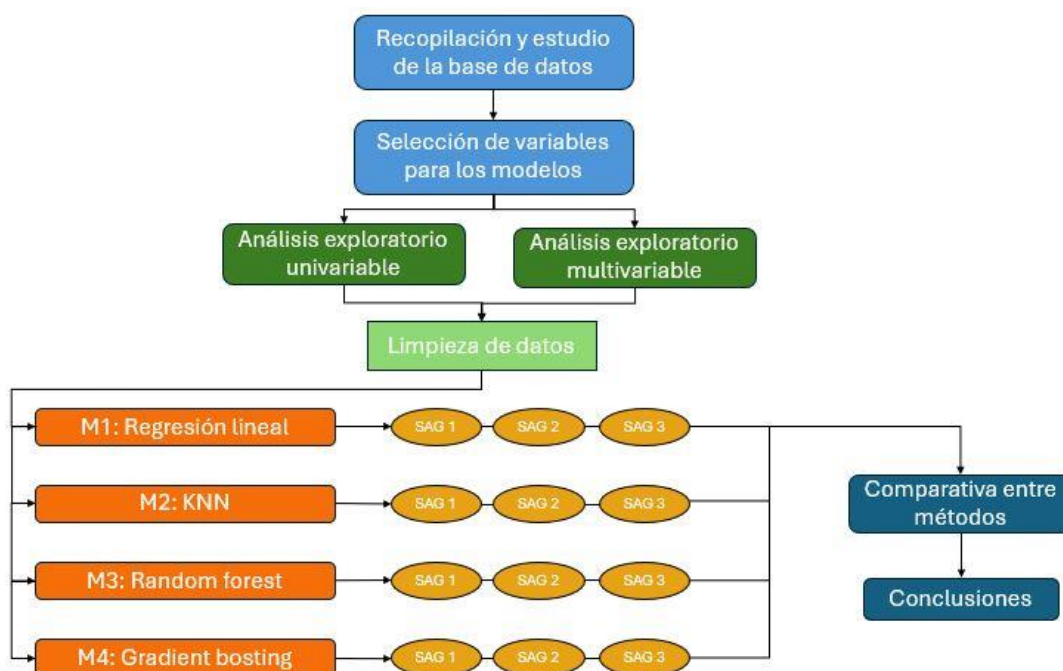


Figura 11: Diagrama resumen de la metodología aplicada en el estudio.

El estudio comenzó con se recopilación y el estudio de la base de datos desde donde se seleccionaron las variables a trabajar y se obtuvo la información para el entrenamiento y testeo de los modelos. Un extracto de esta base de datos se muestra en la Tabla 1 para dar a conocer el formato de esta y las variables que incluye.

Tabla 1: Sección de la base de datos utilizada para el estudio. Incluye variables de dureza, capacidad y granulometría.

		CuT	Fe/FeS2	BWi	DWi	Abx	Tratamiento L1 (ton)	Tratamiento L2 (ton)	Tratamiento L3 (ton)	Tratamiento Planta (ton)	P80 L1	P80 L2	P80 L3	P80 Planta
150:AL_FLOTPRIM.C\FLOTPRIM.FcT_BSD														
01-11-2020 7:00	01-11-2020	1.12	2.73	5.53	12.46	0.21	30040	31500	107851	163331	264	262	259	261
02-11-2020 7:00	02-11-2020	1.06	3.16	5.55	12.48	0.17	28201	30563	107860	166625	255	234	261	255
03-11-2020 7:00	03-11-2020	1.34	5.33	5.40	12.38	0.20	27637	19473	105488	152597	190	182	233	219
04-11-2020 7:00	04-11-2020	1.40	6.38	5.85	12.38	0.17	17037	24149	105971	147158	177	181	232	217
05-11-2020 7:00	05-11-2020	1.42	5.27	6.29	12.84	0.14	30288	30133	109023	163444	221	245	237	235
06-11-2020 7:00	06-11-2020	1.21	5.03	6.75	12.85	0.16	27219	32402	108433	168054	223	233	230	229
07-11-2020 7:00	07-11-2020	1.23	4.65	6.41	12.57	0.17	26306	32144	111182	170232	210	213	242	232
08-11-2020 7:00	08-11-2020	1.14	5.53	6.86	13.25	0.14	26630	33168	111903	171761	211	207	235	226
09-11-2020 7:00	09-11-2020	1.22	5.55	6.77	13.50	0.15	25033	29122	104535	158750	241	247	235	238
10-11-2020 7:00	10-11-2020	0.93	5.86	6.45	13.75	0.13	13746	23484	106240	143470	239	260	261	259
11-11-2020 7:00	11-11-2020	0.88	5.48	5.89	13.33	0.13	21854	21087	110405	153346	257	263	261	261

La frecuencia en las observaciones en la base de datos es diaria e incluye variables de ley de cobre del mineral alimentado, dureza del mineral (BW_i, DW_i, Axb), cantidad de mineral procesado durante el día en cada línea SAG y el P80 promedio de alimentación a cada línea.

Adicionalmente, los datos recopilados incluyen las variables operacionales de cada uno de los tres molinos SAG, entre estas se desatacan: porcentaje de sólidos, velocidad de giro, potencia y consumo de medios de molienda. Una sección completa de todas las variables disponibles en la base de datos se puede ver en las Tablas 2,3 y 4, para los escenarios SAG 01, 02 y 03, respectivamente.

Tabla 2: Sección de la base de datos que muestra las variables operacionales disponibles del SAG 01.

SAG 01														
Tratamiento L1 (tph)	Carga circulante L1	% sólidos L1	Presión en descaños L1 (psi)	%-2" L1	Pebbles generados L1	Veloc. Giro SAG01 (rpm)	Horas de operación L1	Ruido SAG01	Sentido de giro SAG 01	Carguío de bolas L1	Potencia SAG01	Agua Alimentada L1	Consumo Bolas (ton)	Consumo Bolas (g/t)
130VIC11951		140ml1solsp	140pi14302			140ml1rpmcp		140OREJA001	140ML1REVROT	G1.CON.BOLAS	140J103	140FIC1260.MY1	CON.BOLA	CON.BOLAS.E
1332	6.81	76.53	917.00	53.55	2195	8.64	23.85	10.02	REVERSO	10.82	6528.23	420.33	10.82	356.82
1303	6.53	76.30	918.24	55.67	1969	8.25	23.22	10.11	REVERSO	8.66	6242.17	393.32	8.66	304.17
1253	4.75	71.10	926.37	56.50	1378	3.35	23.63	21.04	REVERSO	3.72	7435.34	493.85	3.72	348.36
775	5.01	70.51	674.39	63.68	839	5.46	16.07	6.63	REVERSO	4.65	4041.28	297.33	4.65	270.15
1386	5.62	77.85	911.54	68.82	1805	7.72	24.00	1.72	DIRECTO	7.88	5351.29	394.37	7.88	257.64
1258	6.55	77.75	912.75	65.25	1907	7.91	24.00	3.15	DIRECTO	11.37	5622.28	359.95	11.37	410.89
1234	5.93	77.89	912.64	65.11	1667	7.74	23.71	3.38	DIRECTO	10.61	5514.56	352.22	10.61	393.33
1234	6.56	77.48	912.38	66.00	1873	7.81	23.82	7.82	DIRECTO	10.81	5579.74	354.60	10.81	401.18
1207	10.44	71.44	910.68	69.29	2919	7.97	23.05	10.09	DIRECTO	30.59	5773.62	405.77	30.59	1209.30
655	9.25	88.24	488.48	63.27	1401	4.23	11.79	8.86	REVERSO	3.63	3242.26	203.53	3.63	261.78

Tabla 3: Sección de la base de datos que muestra las variables operacionales disponibles del SAG 02.

SAG 02															
Tratamiento L2 (tph)	Carga circulante L2	% sólidos L2	Presión en decanos L2 (psi)	%-2" L2	Pebbles generados L2	Veloc. Giro SAG02 (rpm)	Horas de Detención L2	Horas de operación L2	Ruido SAG02	Sentido de giro SAG 02	Carguío de bolas L2	Potencia SAG02	Agua Alimentada L2	Consumo Bolas (ton)	Consumo Bolas [g/t]
130VIC21951		140ml2solsp.m	140pi24302			140ml2rpmcp	2.HRS.Oper_BSD		140OREJA002	140ML2REVROT	G2.CON.BOLAS	140J203	140FIC2260.MY2	CON.BOLA	CON.BOLAS.E
1528	8.23	74.97	845.28	63.86	2312	8.32		24.00	0.00	DIRECTO	10.19	6500.54	508.68	10.19	313.87
1278	8.61	73.24	850.10	67.21	2548	7.54		23.40	0.00	DIRECTO	3.56	5792.66	445.52	3.56	353.56
1340	7.92	73.54	847.78	65.60	2457	7.67		24.00	0.00	DIRECTO	8.17	5887.36	480.33	8.17	285.72
1548	8.09	72.52	846.64	66.30	2922	8.24		23.58	0.00	DIRECTO	10.20	6545.26	555.71	10.20	307.14
1371	8.12	72.60	843.73	64.12	2600	8.34		24.00	0.00	DIRECTO	10.18	6734.42	514.54	10.18	346.03
1357	7.85	73.31	841.48	65.26	2458	8.18		24.00	0.00	DIRECTO	3.17	6553.88	493.62	3.17	314.04
1477	5.92	75.39	838.07	55.19	1835	8.31		23.77	0.00	DIRECTO	3.60	7265.54	467.19	3.60	234.18
1558	8.34	76.70	837.34	61.77	3035	8.31		24.00	0.00	DIRECTO	10.60	7086.31	463.88	10.60	317.92
1147	6.75	71.68	835.50	40.85	1808	8.26		21.62	0.00	DIRECTO	8.66	6844.85	369.51	8.66	346.89
1282	7.92	74.45	830.75	47.13	2372	8.87		23.74	0.00	DIRECTO	8.80	7241.18	395.15	8.80	319.24

Tabla 4: Sección de la base de datos que muestra las variables operacionales disponibles del SAG 03.

SAG 101H																	
Tratamiento L3 (tph)	Carga circulante L3	% sólidos L3	Presión en descaños L3 (psi)	%-2" L3	Pebbles generados L3	Veloc. Giro SAG101 (rpm)	Horas de Detención L3	Horas de operación L3	Ruido Norte SAG101	Ruido Sur SAG101	Sentido de giro SAG 101	Carguío de bolas L3	Potencia SAG101	Potencia SAG101	Agua Alimentada L3	Consumo Bolas (ton)	Consumo Bolas (g/t)
141VOT11950B		141SOLIDO101B	141U10101B			141U10101A	3.HRS.Oper_BSD		141AT11008A	141AT11008B	141U10101M	G3.CON.BOLAS	141u10101d		141FIT1002	CON.BOLA	CON.BOLAS.E
2821	6.13	74.31	1404.58	43.41	4087	8.13		23.12	0.00	19.06	DIRECTO	0.00	18626.29	18626.29	768.03	0.00	0.00
3183	6.59	79.53	1406.55	47.79	4362	7.56		24.00	0.00	6.09	DIRECTO	18.17	16666.51	16666.51	818.97	18.17	258.46
3959	7.81	79.20	1410.33	44.64	7125	8.48		24.00	0.00	12.79	DIRECTO	19.62	18987.70	18987.70	1053.90	19.62	226.81
4396	8.21	75.43	1408.56	49.20	9663	8.71		23.37	0.00	18.01	DIRECTO	25.73	19634.51	19634.51	1411.00	25.73	237.69
5216	5.54	76.58	1411.08	51.70	6828	8.94		24.00	0.00	16.62	DIRECTO	29.10	19512.23	19512.23	1422.38	29.10	243.68
4411	5.31	77.92	1411.97	51.07	5536	8.05		24.00	0.00	8.82	DIRECTO	29.77	17216.60	17216.60	1253.26	29.77	301.43
4810	5.34	78.06	1416.22	50.88	6077	8.30		24.00	0.00	8.57	DIRECTO	22.39	18098.50	18098.50	1350.89	22.39	213.51
4424	5.81	78.87	1413.12	55.86	6078	7.63		24.00	0.00	4.75	DIRECTO	32.80	16358.79	16358.79	1164.57	32.80	332.00
4721	5.52	78.01	1414.29	53.98	6166	7.99		24.00	0.00	6.06	DIRECTO	27.31	17425.28	17425.28	1330.24	27.31	258.98
4634	5.03	76.68	1412.96	53.35	5516	8.06		24.00	0.00	8.66	DIRECTO	30.14	17675.19	17675.19	1378.66	30.14	289.64

Posteriormente, se procedió a la selección de variables de interés según los objetivos del estudio. Las variables seleccionadas para el estudio se pueden ver por su correspondiente nombre en la base de datos en la Tabla 5.

Tabla 5: Variables de la base de datos seleccionadas para el estudio y procesamiento.

	SAG 1	SAG 2	SAG 3
Dureza mineral	BWi DWi Abx	BWi DWi Abx	BWi DWi Abx
Granulometria	P80 L1	P80 L2	P80 L3
Porcentaje de sólidos	% sólidos L1	% sólidos L2	% sólidos L3
Velocidad de giro	Veloc. Giro SAG01 (rpm)	Veloc. Giro SAG02 (rpm)	Veloc. Giro SAG1011
Capacidad de procesamiento	Tratamiento L1 (tph)	Tratamiento L2 (tph)	Tratamiento L3 (tph)
Horas de operación	Horas de operación L1	Horas de operación L2	Horas de operación L3
Potencia	Potencia SAG01	Potencia SAG02	Potencia SAG1011

Las variables de dureza fueron estudiadas en la etapa de estudio de correlación (explicada más adelante) donde se observó el comportamiento que presentaban entre ellas y en caso de presentar una correlación directa muy alta solamente se optó por trabajar con la que presentaba mejor calidad de información y mayor cantidad de observaciones válidas. Por otra parte, las variables de “Granulometría”, “Porcentaje de sólidos”, “Tonelaje” y “Velocidad de giro” fueron seleccionadas como variables *input* a los modelos a desarrollar ya que según la literatura presentan una influencia considerable en la energía requerida por los equipos. En cuanto al parámetro “Horas de operación”, fue utilizado como filtro para eliminar observaciones donde el tiempo de operación diario fuera demasiado bajo, lo que indicaba un día de trabajo atípico en la planta y, por lo tanto, mediciones no tan confiables que podían generar perturbaciones importantes en los modelos. Por último, la variable objetivo “Potencia” es la que se estimará a partir de los *inputs* entregados. La base de datos completa de las variables utilizadas no fue adjuntada por su carácter confidencial.

A continuación, se realizó el análisis exploratorio de los datos dividido en dos secciones: univariable y multivariable, en la primera se incluyó todo lo relacionado a la estadística descriptiva de las distintas variables previamente seleccionadas, abarcando medidas de tendencia central, medidas de dispersión, histogramas y gráficos de dispersión frente a la variable objetivo o dependiente. Mientras que en la segunda se buscó estudiar la matriz de correlación entre variables y diagramas de dispersión entre las variables más correlacionadas (Ver Figura 12). Todo este proceso se realizó mediante rutinas de programación en lenguaje

Python y utilizando librerías como *Matplotlib* y *Seaborn*, diseñadas principalmente para el estudio de bases de datos mediante herramientas gráficas.

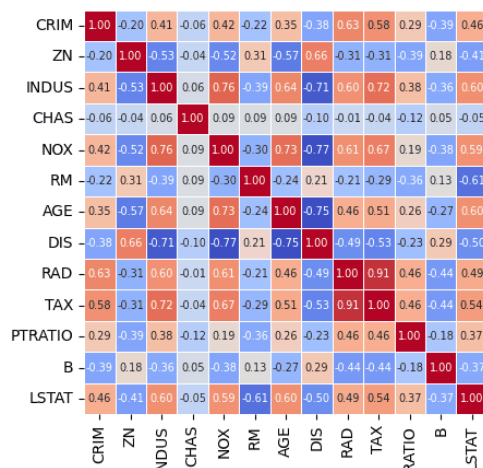


Figura 12: Ejemplo de matriz de correlación en un análisis multivariable.

Luego, se continuó a la etapa de limpieza de datos. Esto, al igual que el análisis exploratorio, se realizó mediante rutinas escritas en Python, pero con ayuda de la librería Pandas, diseñada para la manipulación de bases de datos (ver Anexos sección b). Aquí se consideraron los resultados de la etapa anterior para eliminar observaciones atípicas. Además, se definió cual es la variable de dureza de mineral que se utiliza como regresor para la predicción de la potencia de los equipos.

El núcleo de esta memoria consistió en el desarrollo de los distintos modelos de aprendizaje automático capaces de predecir la potencia. Se procedió a el desarrollo y ajuste de hiperparámetros para cuatro distintos algoritmos: Regresión lineal múltiple, *K-Nearest neighbors*, *Random forest* y *Gradient boosting*, cada uno se aplicó a la base de datos propia para cada una de las tres líneas de molienda SAG, lo que finalmente entregó un total de doce modelos distintos desarrollados, los cuales posteriormente fueron evaluados mediante sus indicadores de error y precisión. Lo anterior se realizó utilizando la librería *Scikit-learn* para Python (Ver Anexos sección b).

Finalmente, los resultados de todos los modelos fueron comparados y se estudió qué algoritmo entregó mejores resultados para cada línea de procesamiento.

CAPÍTULO IV: RESULTADOS Y ANÁLISIS

4.1 ANÁLISIS EXPLORATORIO UNIVARIABLE Y LIMPIEZA DE DATOS

Mediante estadística descriptiva se buscaba conocer mejor la muestra de los datos en estudio. En primer lugar, se calculó: cantidad de datos, media aritmética, desviación estándar, mínimo, máximo, primer cuartil, segundo cuartil y tercer cuartil. Lo anterior para las variables seleccionadas y en tres bases de datos distintas, una para cada línea de procesamiento. De esta forma se obtuvieron tres tablas (Ver Tablas 6, 7 y 8).

Tabla 6: Estadística descriptiva de las variables seleccionadas para la base de datos de la línea SAG 1.

	BWi (kWh/t)	Dwi (kWh/m ³)	Axb	Tratamiento L1 (ton)	P80 L1 (um)	% sólidos L1	Veloc. Giro SAG01 (rpm)	Horas de operación L1	Potencia SAG01 (kW)
observaciones	585	585	585	589	589	589	589	589	589
media	6.26	12.50	0.18	28963.5	236	75.5	8.21	22.55	6216.37
desv. est.	0.57	0.54	0.05	6635.1	51	5.2	1.79	4.71	1420.52
min	4.42	9.82	0.10	0.0	0	47.9	0.00	0.00	0.00
p25	5.86	12.12	0.16	28058.3	224	73.2	8.18	23.66	6020.08
p50	6.28	12.44	0.18	30456.4	247	74.5	8.64	23.95	6489.06
p75	6.66	12.85	0.20	32339.9	263	76.1	9.05	24.00	6940.43
max	8.30	14.04	0.81	37467.4	308	100.0	9.72	24.00	7874.20

Tabla 7: Estadística descriptiva de las variables seleccionadas para la base de datos de la línea SAG 2.

	BWi (kWh/t)	Dwi (kWh/m ³)	Axb	Tratamiento L1 (ton)	P80 L1 (um)	% sólidos L1	Veloc. Giro SAG01 (rpm)	Horas de operación L1	Potencia SAG01 (kW)
observaciones	585	585	585	589	589	589	589	589	589
media	6.26	12.50	0.18	28701.0	233	75.7	8.10	22.55	6348.06
desv. est.	0.57	0.54	0.05	6894.6	53	4.9	1.83	4.95	1473.39
min	4.42	9.82	0.10	0.0	0	67.4	0.00	0.00	0.00
p25	5.86	12.12	0.16	27939.7	222	73.8	8.07	23.68	6178.56
p50	6.28	12.44	0.18	30370.5	243	74.8	8.53	24.00	6698.34
p75	6.66	12.85	0.20	32248.4	262	76.0	8.92	24.00	7067.57
max	8.30	14.04	0.81	37117.0	304	100.0	9.87	24.00	7851.89

Tabla 8: Estadística descriptiva de las variables seleccionadas para la base de datos de la línea SAG 3.

	BWi (kWh/t)	Dwi (kWh/m3)	Axb	Tratamiento L1 (ton)	P80 L1 (um)	% sólidos L1	Veloc. Giro SAG01 (rpm)	Horas de operación L1	Potencia SAG01 (kW)
observaciones	585	585	585	589	589	589	589	589	589
media	6.26	12.50	0.18	94694.8	231	71.3	8.21	22.73	20104.62
desv. est.	0.57	0.54	0.05	23691.5	55	15.9	1.93	4.91	15749.78
min	4.42	9.82	0.10	0.0	0	0.0	0.00	0.00	0.00
p25	5.86	12.12	0.16	94770.7	223	74.3	8.31	24.00	18823.00
p50	6.28	12.44	0.18	101572.0	242	75.3	8.69	24.00	20218.38
p75	6.66	12.85	0.20	106191.5	259	76.2	9.01	24.00	21287.59
max	8.30	14.04	0.81	117103.2	315	81.8	9.66	24.00	327785.17

Adicionalmente, se crearon histogramas de los mismos datos ya representados en las Tablas 6, 7 y 8, esto con el fin de observar la distribución de cada variable (BWi, DWi, Axb, tonelaje procesado, P80 promedio y porcentaje de sólidos). Se generaron tres sets de histogramas: para las bases de datos de las líneas SAG01, SAG02 y SAG03 (Ver Figuras 13, 14 y 15, respectivamente). En todos los escenarios se puede ver que son pocas las variables con una distribución cercana a la normal. También se observa la clara existencia de datos atípicos en las barras alejadas del resto de las observaciones.

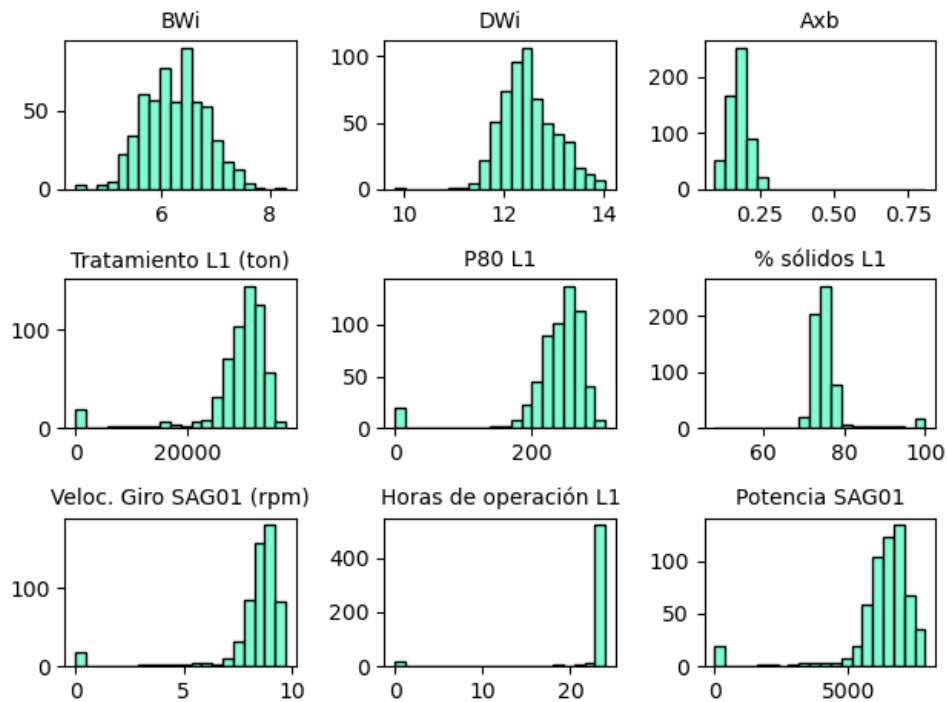


Figura 13: Set de histogramas generados para las variables de la línea SAG 1 sin limpieza de datos.

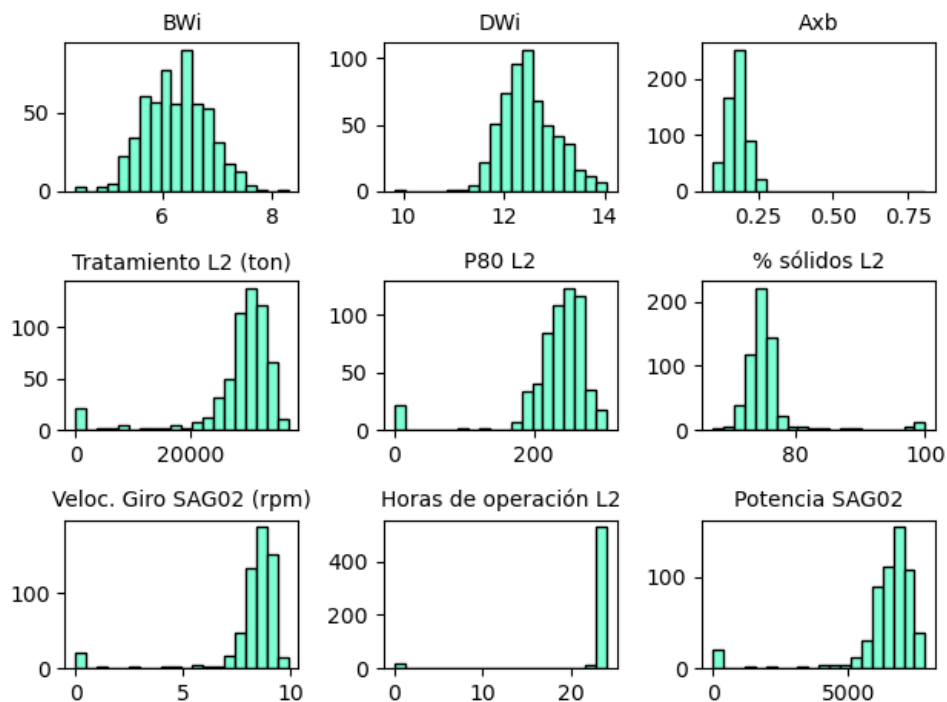


Figura 14: Set de histogramas generados para las variables de la línea SAG 2 sin limpieza de datos.

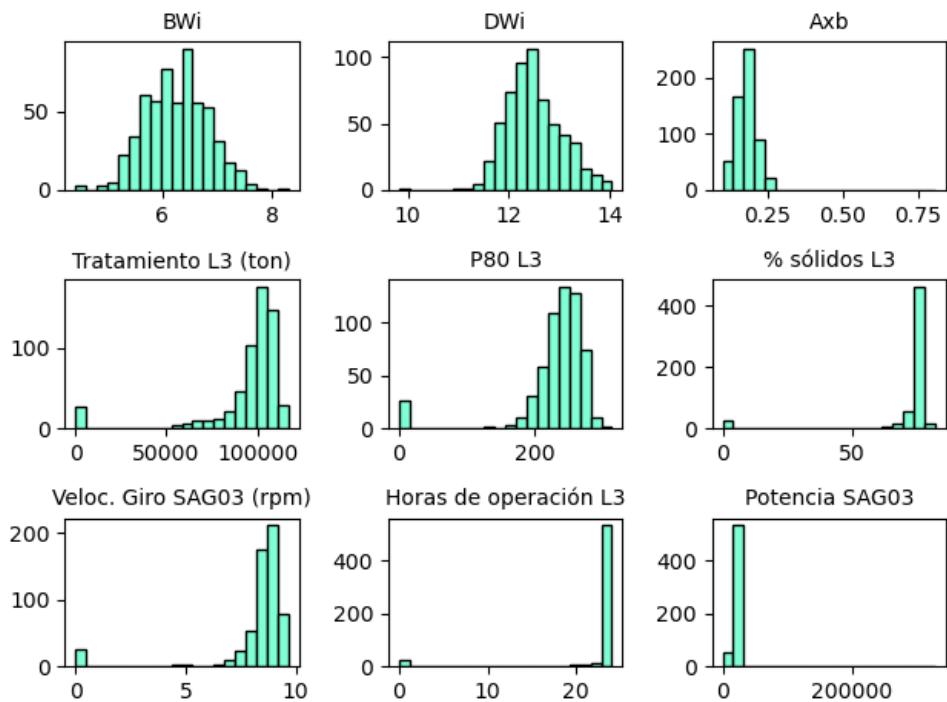


Figura 15: Set de histogramas generados para las variables de la línea SAG 3 sin limpieza de datos.

Las medidas de tendencia central, medidas de dispersión e histogramas, en conjunto, permiten una mejor limpieza de datos. Se procedió a analizar cada variable para eliminar datos atípicos, por ejemplo, para las tres líneas se pueden observar mínimos de 0 en la variable “Horas de operación”, lo que indica que para estas observaciones la respectiva línea de procesamiento tuvo detenciones (programadas o no programadas) o simplemente no fue alimentada con mineral proveniente del *stockpile*. Así mismo, también existen observaciones donde datos de dureza exceden en más de tres veces la desviación estándar calculada, lo que en una distribución normal estándar excede el 99,7% de los datos esperables. Esto se puede apreciar claramente en los histogramas de la variable, donde hay valores que se escapan bastante del resto de la distribución. Estos probables valores atípicos se pueden explicar por fallos de medición de los sensores utilizados para captar estos atributos.

Para buscar y eliminar valores atípicos se utilizó un método de corte por percentiles para los valores más altos y bajos, estos fueron ajustados según el criterio del analista de los datos al observar los histogramas generados para cada variable en múltiples iteraciones. Inicialmente las tres bases de datos contenían 585 observaciones con todas sus variables no nulas. Luego de la limpieza estas contenían las cantidades indicadas en la Tabla 9.

Tabla 9: Cantidad inicial y final de observaciones para cada base de datos.

	SAG 1	SAG 2	SAG 3
Observaciones iniciales	585	585	585
Observaciones finales	489	500	468
Datos conservados	83,6%	85,5%	80,0%

Como se puede observar, el porcentaje de datos conservados para ningún conjunto fue inferior al 80%, buscando un equilibrio entre cantidad y calidad de la información a utilizar. Por otro lado, los parámetros de corte que se utilizaron finalmente se muestran en las Tablas 10, 11 y 12.

Tabla 10: Percentiles de corte inferior y superior para las variables en la base de datos de la línea SAG01.

		Percentil inferior	Percentil superior
SAG 01	BWi	1	99
	DWi	1	99
	Axb	1	99
	P80 [um]	3	99
	% sólidos	1	98
	Tratamiento [ton]	6	-
	Velocidad de giro [rpm]	7	-
	Horas de operación	9	-
	Potencia molino [kW]	3	99

Tabla 11: Percentiles de corte inferior y superior para las variables en la base de datos de la línea SAG02.

		Percentil inferior	Percentil superior
SAG 02	BWi	1	99
	DWi	1	99
	Axb	1	99
	P80 [um]	4	99
	% sólidos	1	98
	Tratamiento [ton]	6	-
	Velocidad de giro [rpm]	6	-
	Horas de operación	6	-
	Potencia molino [kW]	3	99

Tabla 12: Percentiles de corte inferior y superior para las variables en la base de datos de la línea SAG03.

		Percentil inferior	Percentil superior
SAG 03	BWi	1	99
	DWi	1	99
	Axb	1	99
	P80 [um]	3	99
	% sólidos	5	99
	Tratamiento [ton]	9	-
	Velocidad de giro [rpm]	7	-
	Horas de operación	8	-
	Potencia molino [kW]	6	98

Para los tres conjuntos los valores de dureza de mineral son los mismos ya que los datos originales asignan el mismo tipo de mineral para las tres líneas de procesamiento, por lo tanto, se usan los mismos valores de corte para su limpieza. Por el contrario, los percentiles de corte para resto de las variables cambian según la distribución de datos iniciales. Además, se define que “Tratamiento”, “Velocidad de giro” y “Horas de operación” sólo serán acotadas en su margen inferior, debido a que las observaciones superiores poseen valores esperados para sus respectivas variables.

Posterior a la limpieza realizada, se obtuvieron los tres nuevos conjuntos de datos, a partir de los cuales nuevamente se generan histogramas para cada variable (Ver Figuras 16, 17 y 18), esto con el fin de observar las nuevas distribuciones y verificar si la información es suficientemente apta para la realización de modelos.

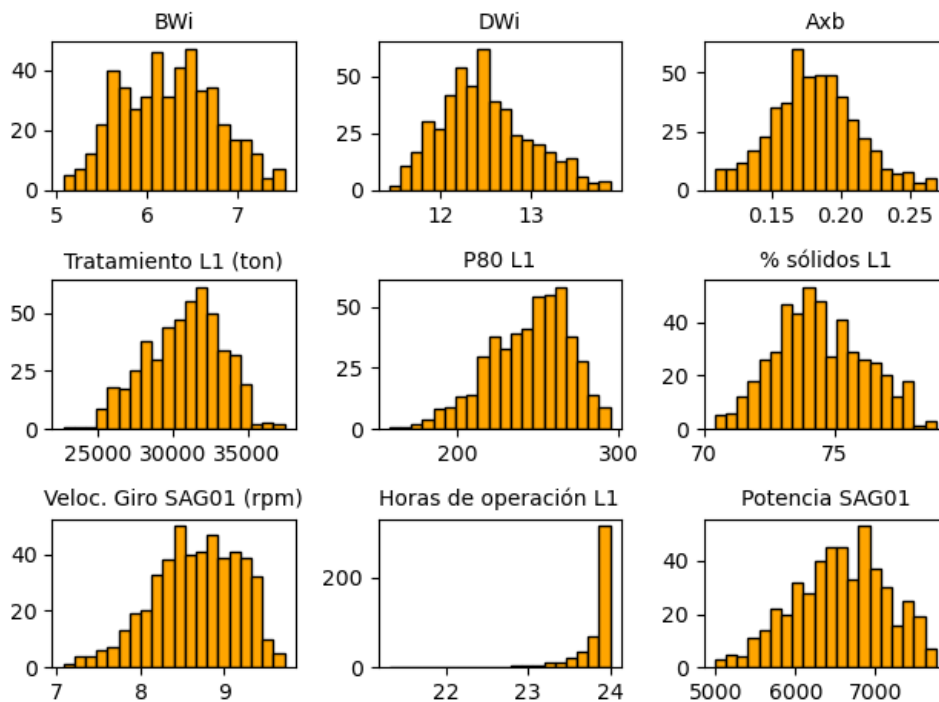


Figura 16: Set de histogramas generados para las variables de la línea SAG 1 tras limpieza de datos.

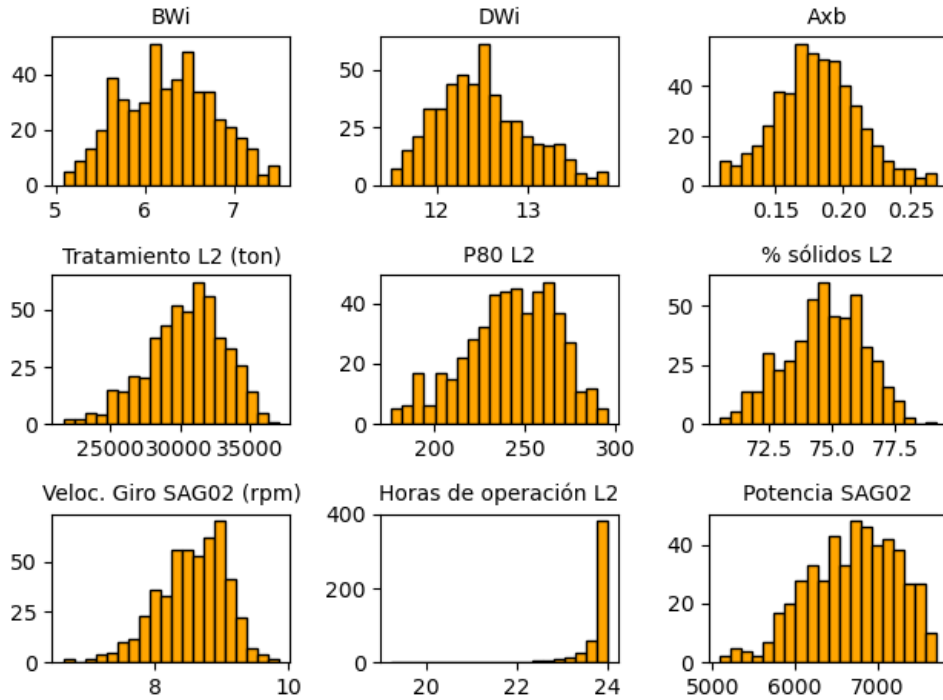


Figura 17: Set de histogramas generados para las variables de la línea SAG 2 tras limpieza de datos.

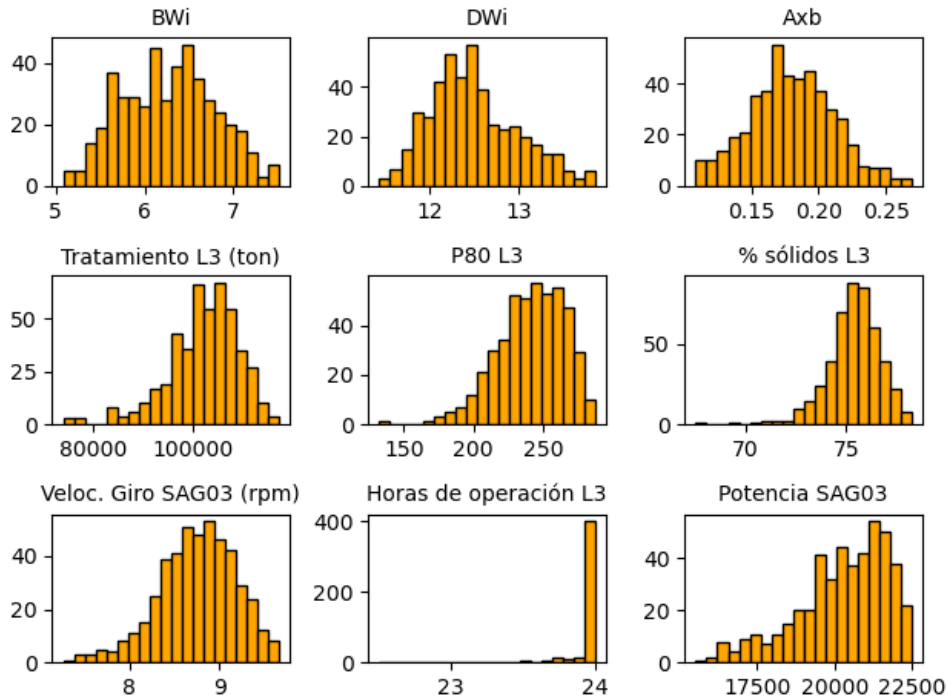


Figura 18: Set de histogramas generados para las variables de la línea SAG 3 tras limpieza de datos.

Se observó de mejor forma que los valores para las variables de dureza (BW_i, DW_i, Axb) presentaban indicios de distribuciones normales, sin embargo, para poder afirmar esto con mayor precisión se necesitaría disponer de una mayor densidad de observaciones. Adicionalmente, se pudo ver que ya no existían valores extremadamente dispersos que pudieran indicar valores atípicos para alguno de los tres escenarios.

4.2 ANÁLISIS EXPLORATORIO MULTIVARIABLE

Por simplicidad, para la realización del análisis exploratorio de dos o más variables en conjunto, se procedió a trabajar con cada escenario por separado. Para cada uno se generó un mapa de calor entre variables, el cual muestra la matriz de correlación de forma gráfica. Adicionalmente, se analizaron múltiples gráficos de dispersión entre pares de variables, para observar la relación existente entre estas y estudiar su posible influencia al utilizarlas como regresores en los modelos de predicción.

Cabe destacar que los valores de correlación entre las tres variables de dureza de mineral estudiadas son los mismos en los tres mapas de calor ya que el mineral alimentado a las tres líneas de molienda es el mismo según la base de datos general.

4.2.1 Indicadores de dureza de mineral

Se realizaron gráficos de dispersión entre las tres distintas variables que explican la moliendabilidad del mineral (Ver Figuras 19, 20 y 21), esto con el fin de observar de forma más específica la correlación que puedan presentar y así poder de seleccionar la más adecuada para el entrenamiento de los modelos. Es de vital importancia el uso de una única variable para este atributo, ya que ingresar dos o más *inputs* que entreguen información similar o muy correlacionada al modelo puede perjudicar los resultados.

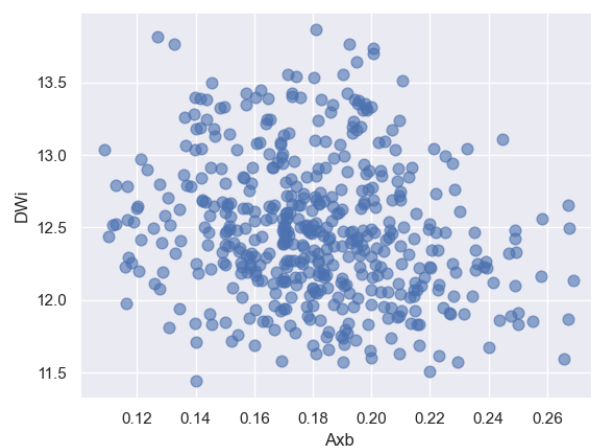


Figura 19: Gráfico de dispersión para medidas de DW_i y Axb .

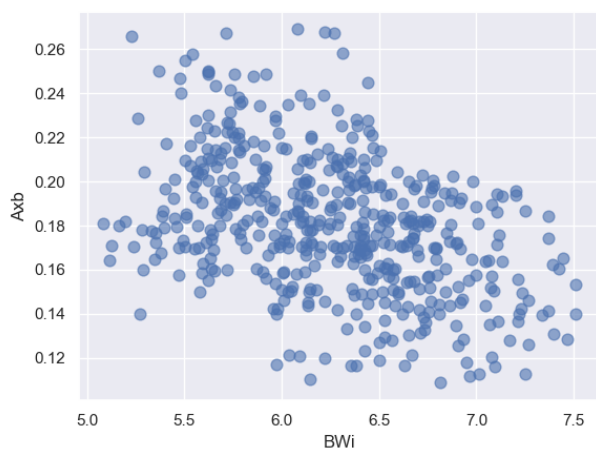


Figura 20: Gráfico de dispersión para medidas de Axb y BW_i .

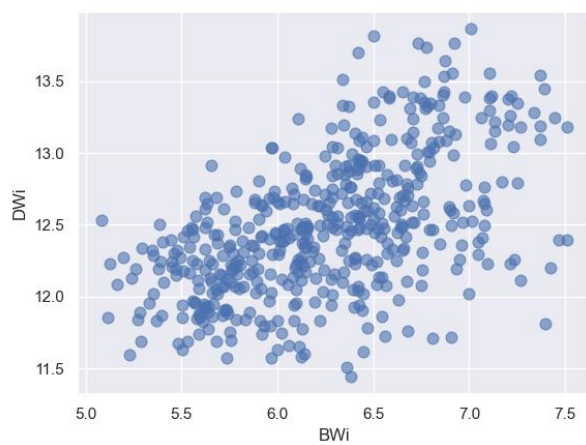


Figura 21: Gráfico de dispersión para medidas de DW_i y BW_i .

Con esta información y con los valores de R^2 mostrados más adelante en las matrices de correlación entre variables (Véase las secciones 8.2, 8.3 y 8.4) se pudo interpretar que la información física del mineral que entregan DWi y BWi es la más concordante entre pares de indicadores, mientras que Abx que presenta los valores de correlación más bajos (-0,19 y -0,42 con DWi y BWi, respectivamente), lo que podría señalar que es el ensayo menos confiable para este estudio. Por otra parte, BWi presentó valores de R^2 mayores respecto a sus pares que DWi. Lo anterior, junto con la amplia utilización del indicador de Bond en la industria y los resultados de los gráficos de indicadores de dureza vs potencia mostrados más adelante, llevaron al analista de datos a la decisión de seleccionar este atributo (BWi) como *input* de los modelos desarrollados.

4.2.2 SAG 01

En el primer escenario, el mapa de calor (Figura 22) muestra que el atributo más correlacionado con la variable objetivo, es decir, la potencia del molino, es la velocidad de giro ($R^2 = 0,67$), esto es esperable ya que están directamente relacionadas en la operación del equipo, por lo tanto, se espera que sea de las más relevantes al momento de predecir el comportamiento del proceso mediante los distintos modelos.

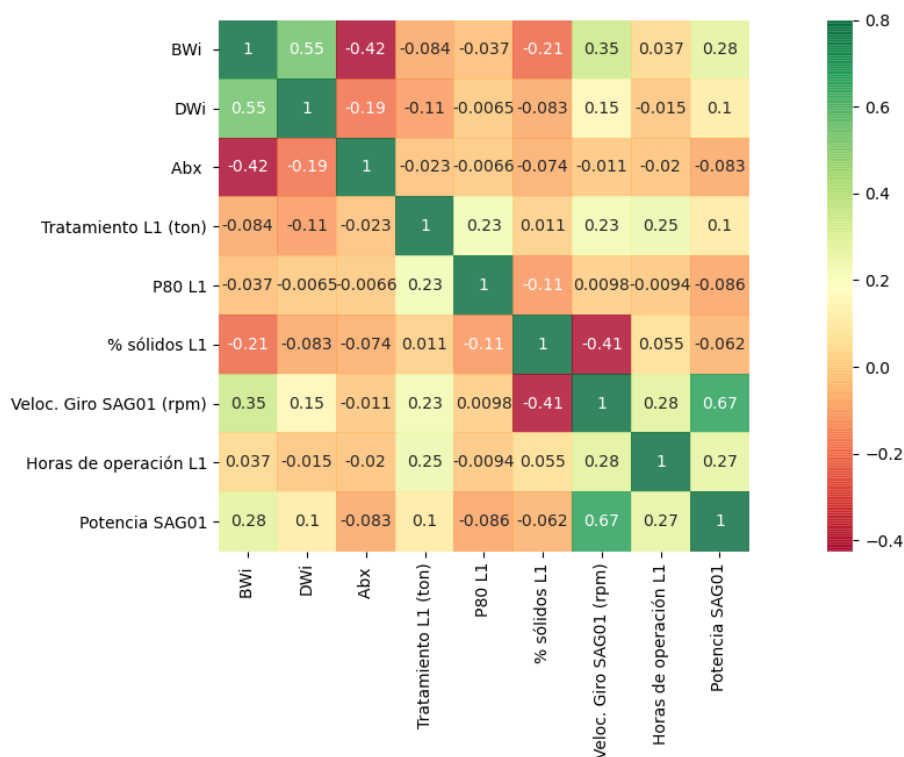


Figura 22: Mapa de calor con valores de correlación entre variables en el primer escenario.

En cuanto a los distintos indicadores de conminución, se puede ver la relación que tienen respecto a la variable objetivo en las Figuras 23, 24 y 25.

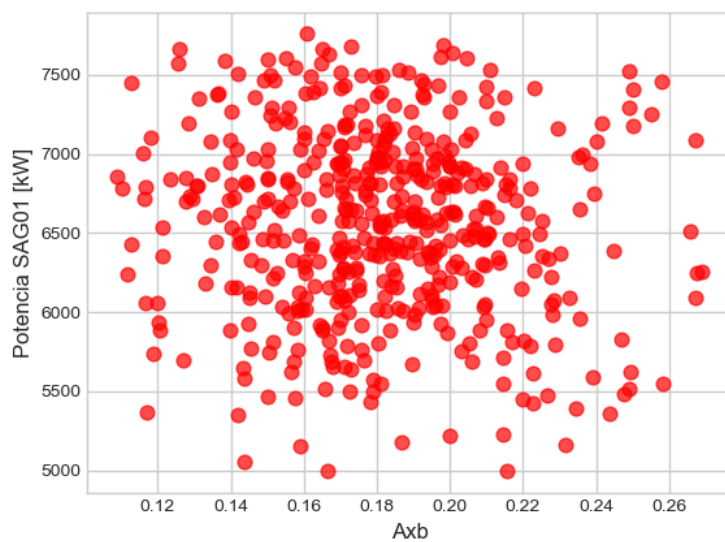


Figura 23: Gráfico de dispersión entre A_{xb} y la potencia del molino para el escenario 1.

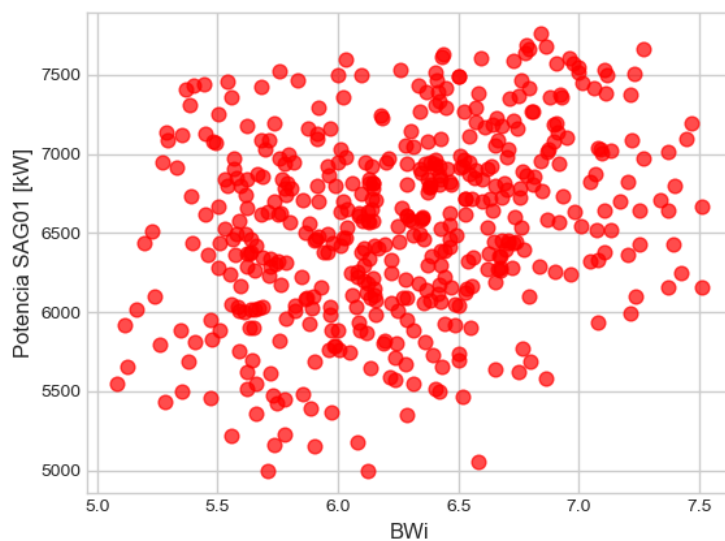


Figura 24: Gráfico de dispersión entre BW_i y la potencia del molino para el escenario 1.

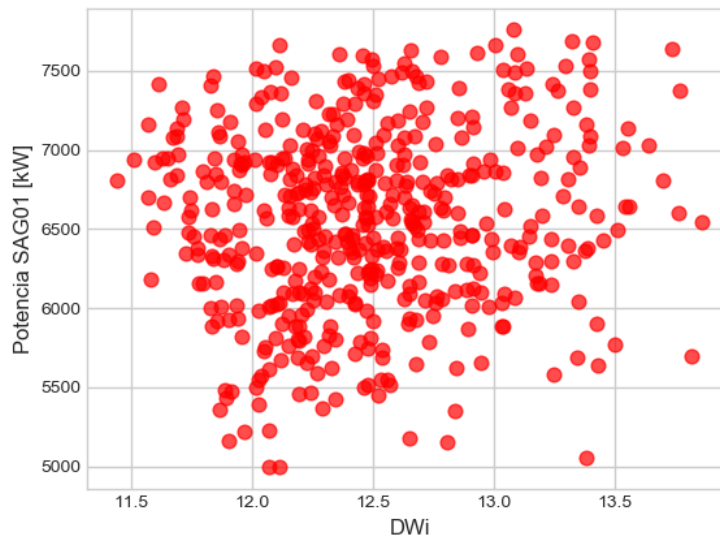


Figura 25: Gráfico de dispersión entre DWi y la potencia del molino para el escenario 1.

Se aprecia que el indicador que tiene una relación más lineal con respecto a la potencia es el BWi y que por otra parte el Axb no muestra un comportamiento con un patrón claro en relación a la misma variable (Ver Figura 23). Esto a su vez se confirma con los valores de correlación del mapa de calor (Figura 22) que son: 0,28, 0,10 y -0,083 para BWi, DWi y Axb, respectivamente.

Adicionalmente, se generaron gráficos de distribución para los valores de las variables operacionales disponibles en la base de datos general con respecto a los valores de la variable dependiente “Potencia del molino” (Ver Figura 26). La relación más directa observable es la de la velocidad de giro del equipo, que como ya se mencionó, es la que posee mayor correlación. También se observa que la variable que representa las horas de operación prácticamente no está relacionada con la potencia, debido a que un gran porcentaje de los datos converge a un único valor (24 horas) y la potencia es un valor promedio medido durante las horas que se utilizó el molino. Por lo anterior, esta variable solo se utilizó como filtro al momento de la limpieza de datos.

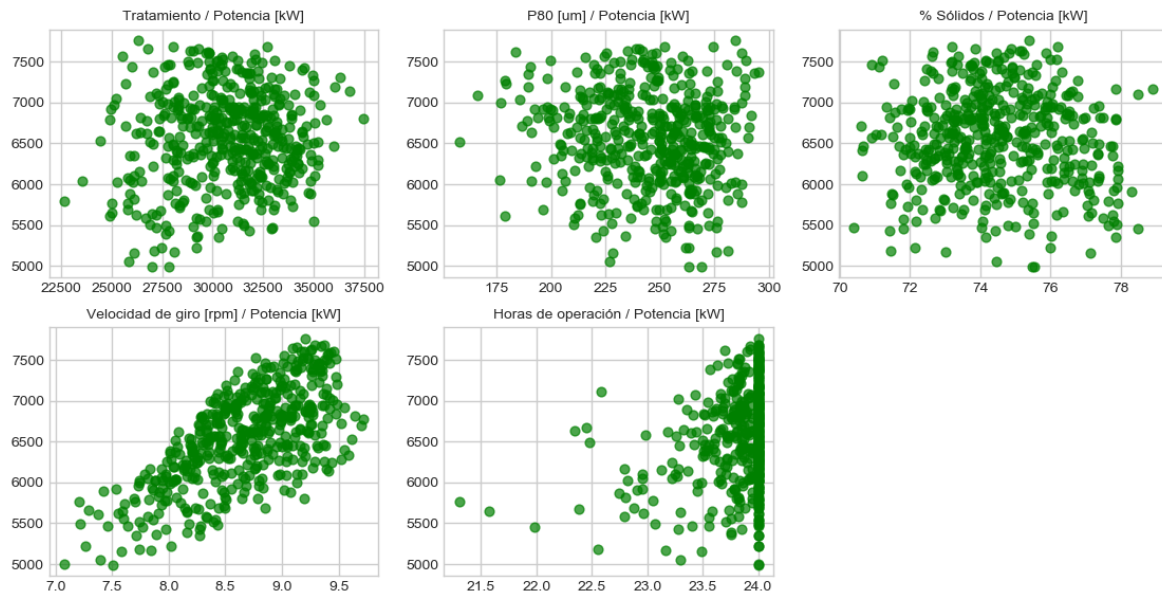


Figura 26: Gráficos de dispersión entre las distintas variables operacionales estudiadas y la potencia del molino para el escenario 1.

4.2.3 SAG 02

Para el caso del segundo escenario, el mapa de calor (Ver Figura 27) muestra algo similar a lo que sucede con los datos de la primera línea de molienda: la velocidad de giro es la variable más correlacionada con la potencia ($R^2 = 0,66$), atribuyéndole la misma explicación que el escenario anterior. Aquí sin embargo se puede ver que el P80 presenta un R^2 un poco más significativo y negativo, por lo que podría esperarse una mayor significancia para los modelos.

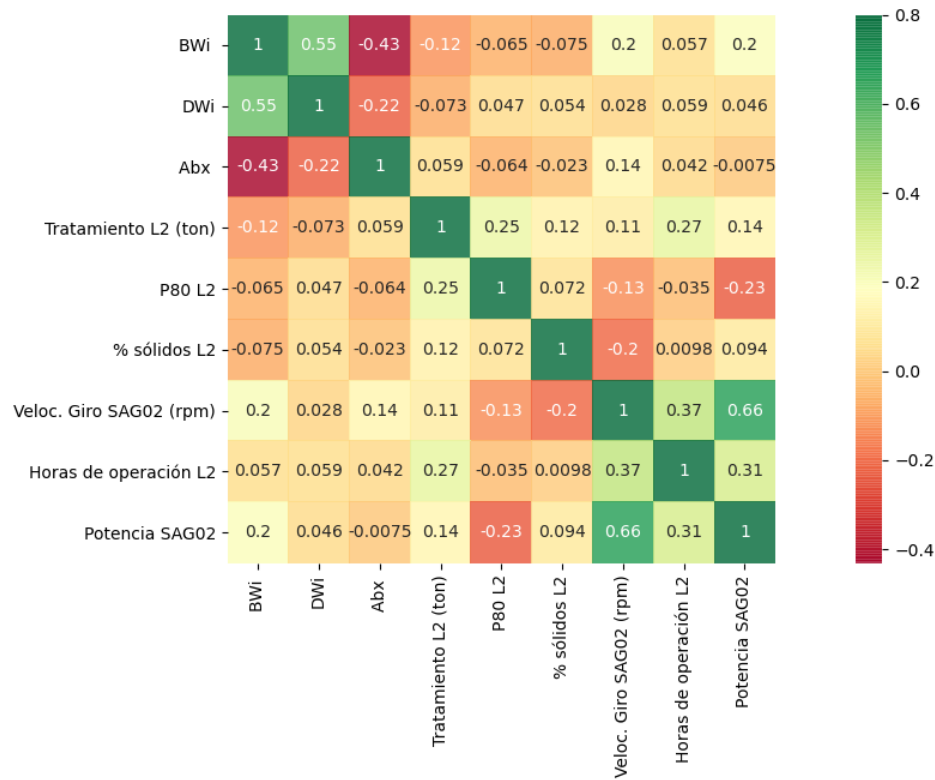


Figura 27: Mapa de calor con valores de correlación entre variables en el segundo escenario.

De igual forma se pueden ver las relaciones que tienen los tres indicadores de conminución respecto a la variable objetivo en las Figuras 28, 29 y 30.

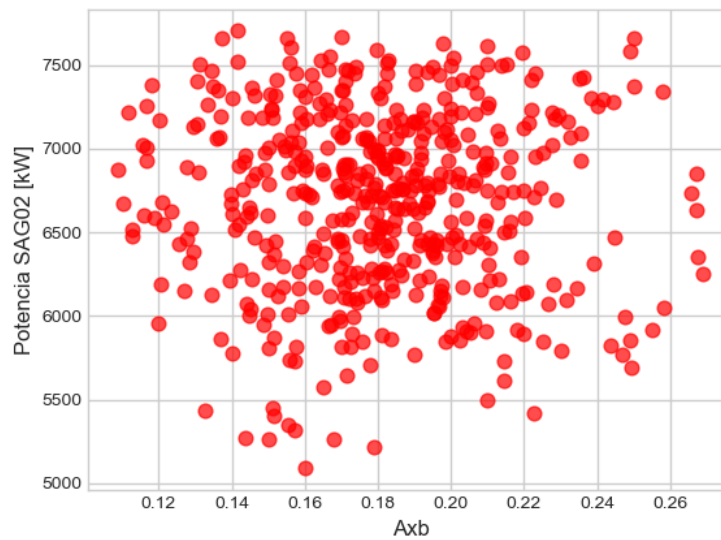


Figura 28: Gráfico de dispersión entre Axh y la potencia del molino para el escenario 2.

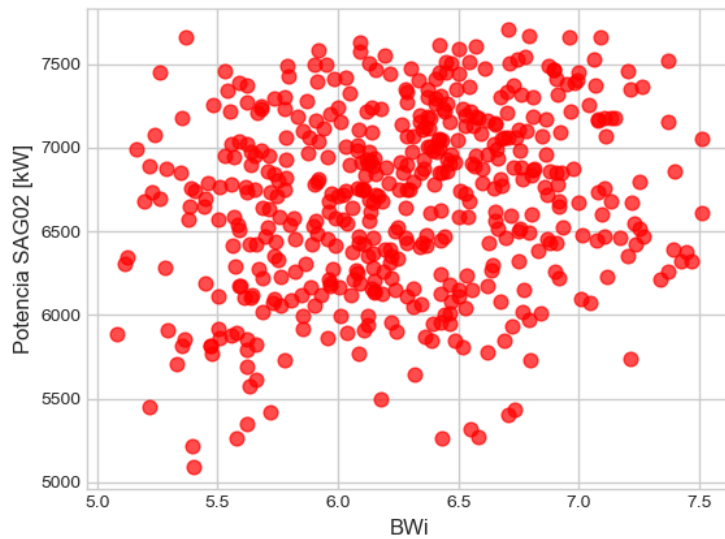


Figura 29: Gráfico de dispersión entre BWi y la potencia del molino para el escenario 2.

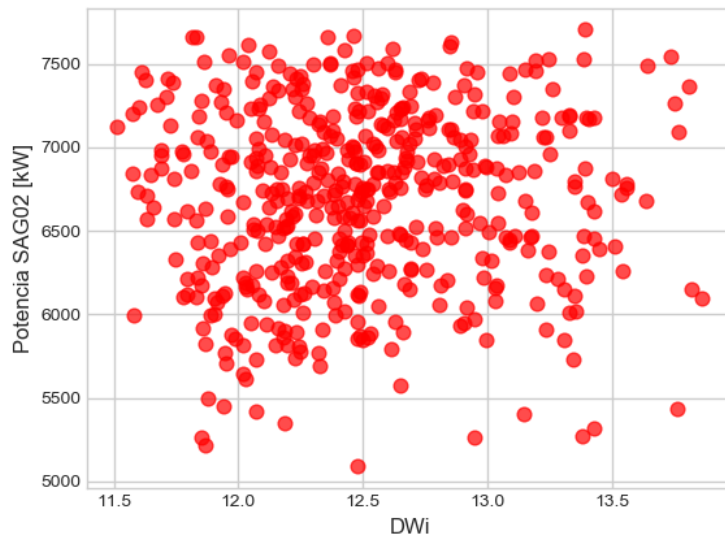


Figura 30: Gráfico de dispersión entre DWi y la potencia del molino para el escenario 2.

En este caso también se pudo apreciar que el indicador que tiene una relación más cercana a ser lineal con respecto a la potencia es el BWi (Ver Figura 29). Los valores de correlación de los indicadores de moliendabilidad en el mapa de calor (Figura 27) son: 0,20, 0,046 y -0,0075 para BWi, DWi y Axb, respectivamente, siendo en general menores que para la línea uno de molienda y entregando un valor casi nulo para el Axb, es decir, esta variable explica menos de un 1% de la varianza en la potencia del molino.

De igual forma, se generaron gráficos de distribución para los valores de las variables operacionales disponibles en la base de datos general con respecto a los valores de la variable dependiente, es decir, potencia del molino (Ver Figura 31). La relación más directa observable es la de la velocidad de giro del equipo, que también posee mayor correlación. En cuanto a la variable “horas de operación” ocurre lo mismo que para el primer escenario, es decir, no se considera como *input* de los modelos al momento de desarrollarlos y sólo se utilizó como filtro en la etapa de limpieza de datos. Por último, como ya se mencionó, se observó una relación más clara en la distribución de los datos en el P80, asemejándose un poco más a una ecuación lineal de pendiente negativa.

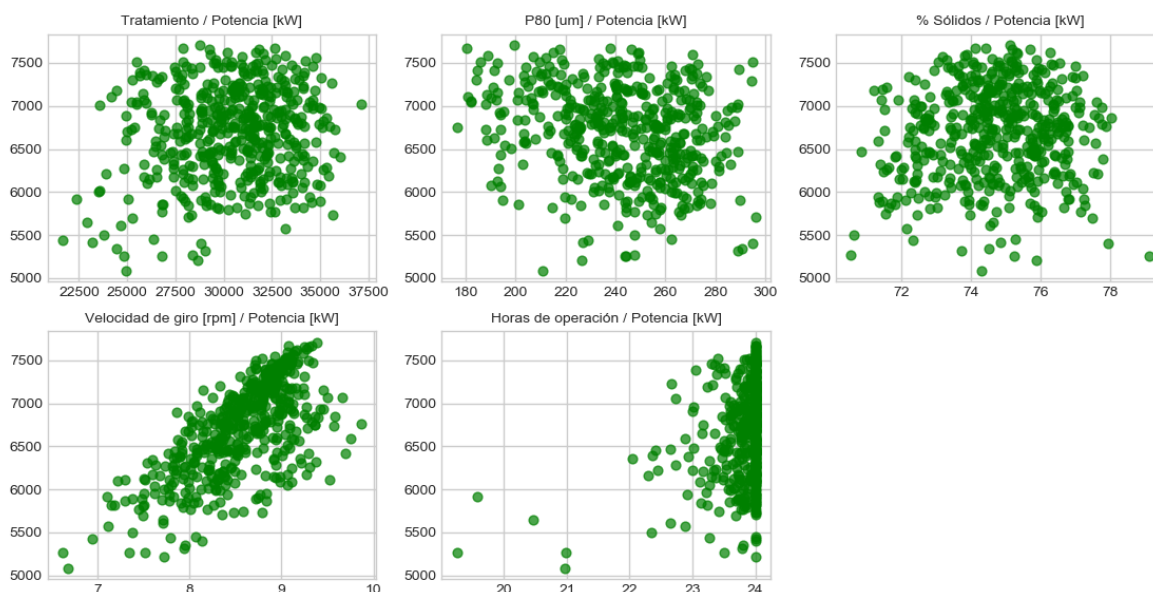


Figura 31: Gráficos de dispersión entre las distintas variables operacionales estudiadas y la potencia del molino para el escenario 2.

4.2.4 SAG 03

Por último, para el escenario de la tercera línea de molienda, el mapa de calor (Ver Figura 32) mostró, en general, correlaciones más altas que en los dos escenarios anteriores y la velocidad de giro sigue siendo la variable con mayor R^2 respecto a la potencia ($R^2 = 0,66$), atribuyéndole la misma explicación que en los escenarios anteriores. Debido a lo que se menciona en un comienzo, se podrían haber esperado mejores resultados en cuanto a las métricas de errores en los distintos modelos desarrollados, ya que las variables parecen explicar mejor la varianza de la variable objetivo.

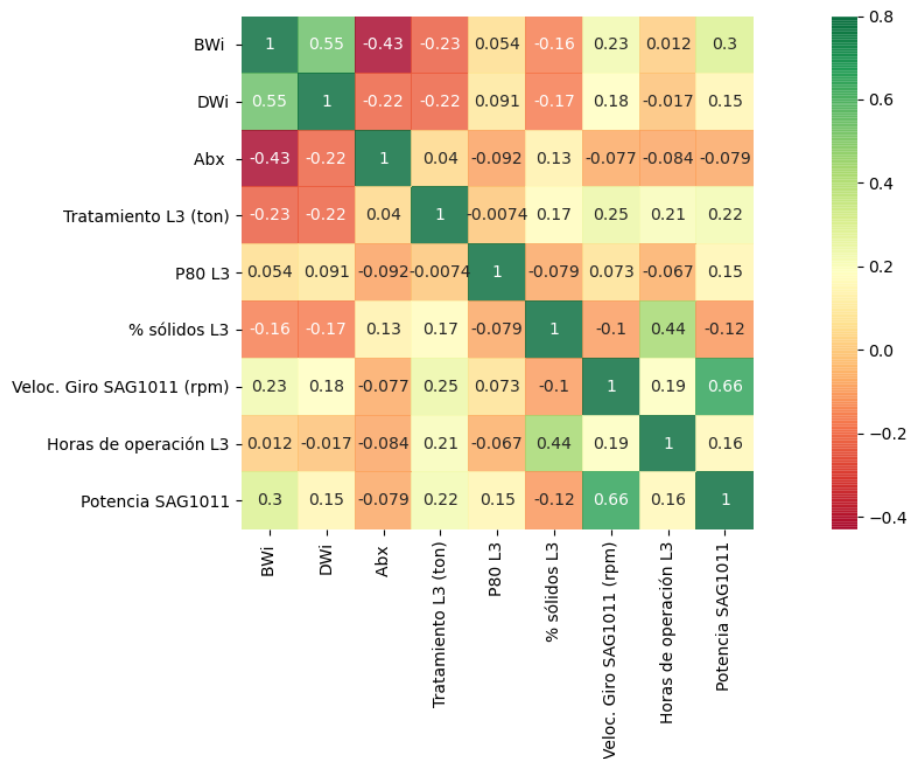


Figura 32: Mapa de calor con valores de correlación entre variables en el tercer escenario.

Al igual que en los dos escenarios anteriores se crearon gráficos de dispersión para ver el comportamiento de la potencia frente a los tres indicadores de moliendabilidad. Estos se presentan en las Figuras 33, 34 y 35.

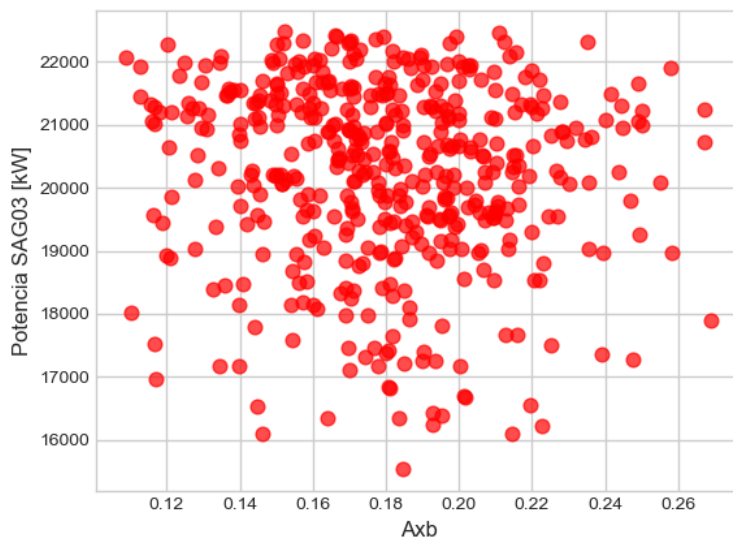


Figura 33: Gráfico de dispersión entre Abx y la potencia del molino para el escenario 3.

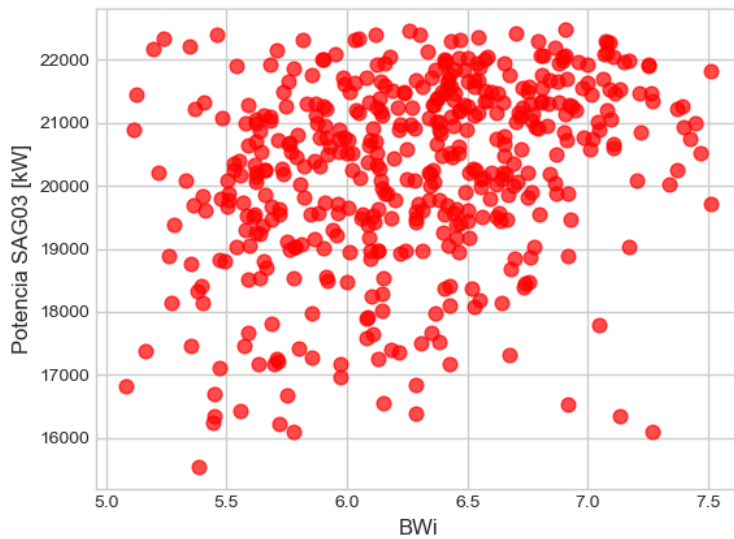


Figura 34: Gráfico de dispersión entre BWi y la potencia del molino para el escenario 3.

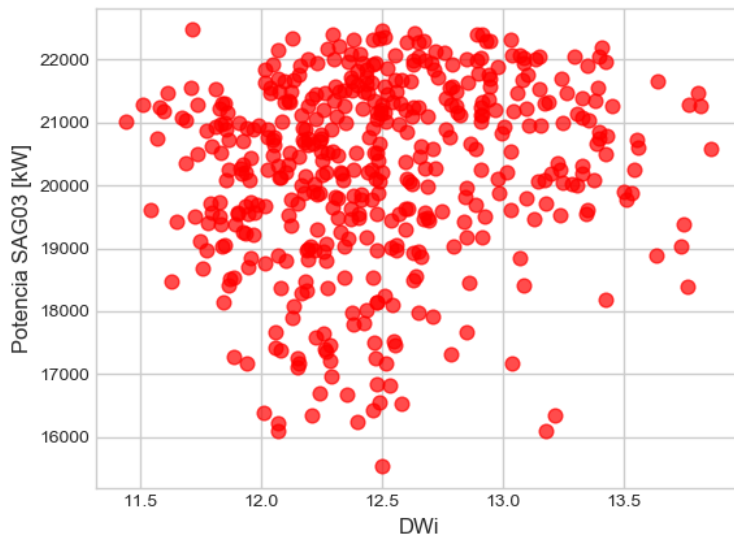


Figura 35: Gráfico de dispersión entre DWi y la potencia del molino para el escenario 3.

En el tercer escenario se observó que los puntos del gráfico DWi y de mayor forma los del gráfico BWi, presentan una especie de comportamiento logarítmico (Ver Figuras 34 y 35), el cual probablemente se distinguiría mejor con una mayor densidad de observaciones. Por el contrario, Axb no presenta alguna tendencia notable. Los valores de correlación de los indicadores de moliendabilidad en el mapa de calor (Figura 32) son: 0,30, 0,15 y -0,079 para BWi, DWi y Axb, respectivamente.

Al igual que en los casos anteriores, la tendencia más clara es la de la variable independiente con la correlación más alta (velocidad de giro del molino), que presenta un comportamiento que tiende a la linealidad. También se pudo apreciar cierta tendencia lineal de pendiente negativa en el gráfico de porcentaje de sólidos (Ver Figura 36), lo que podría indicar que en este equipo las pulpas más densas tendrían cierto efecto en la potencia consumida por el molino, reduciéndola. Esta hipótesis requeriría una mayor cantidad de observaciones para estudiarse mejor, ya que el rango en el que se mueven la mayoría de los valores de esta variable es bastante acotado (72% - 78%) y esto puede inducir error.

Finalmente, la situación con la variable de horas de operación del molino es igual a la de los dos escenarios anteriores, no fue considerada como *input* en los modelos por las razones ya explicadas.

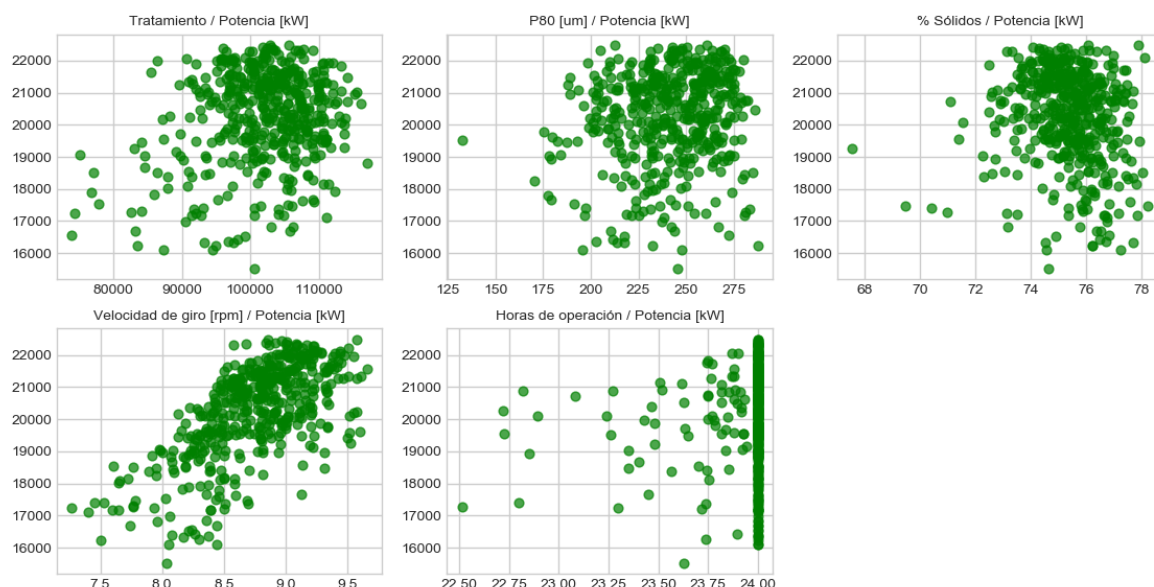


Figura 36: Gráficos de dispersión entre las distintas variables operacionales estudiadas y la potencia del molino para el escenario 3.

4.3 DESARROLLO Y EVALUACIÓN DE MODELOS

4.3.1 Método 1: Regresión lineal

Para el desarrollo de los modelos utilizando el algoritmo de regresión lineal múltiple (Ver sección 2.2.1) se consideraron como atributos: “BWi (kWh/t)”, “Tratamiento (tph)”, “P80

(μm)”, “Porcentaje de sólidos”, “Velocidad de giro (rpm)”. La variable objetivo, al igual que en el resto de los algoritmos utilizados, es “Potencia (kW)”.

Estos modelos fueron desarrollados y evaluados mediante el método de validación cruzada *Repeated K-fold* (Ver sección 2.3.2.3), definiendo la cantidad de conjuntos en tres y realizando cien iteraciones aleatorias con distintos conjuntos generados. Luego se calculó el error promedio de todas las iteraciones, mediante el cálculo del *Mean Squared Error* (MSE) y *Root Mean Squared Error* (RMSE) (Ver sección 2.3.1). Adicionalmente, se obtuvieron los coeficientes para cada variable independiente del modelo, el intercepto y el R^2 de los valores predichos respecto a los valores reales.

4.3.1.1 SAG 01

Tras realizar la validación cruzada utilizando los datos asignados para el escenario de la primera línea de molienda, se obtuvo que la ecuación del modelo es la siguiente:

$$\text{Potencia} = -7.601 + 35,47BW_i - 0,02475 \text{ Tratamiento} - 1,058 P80 + 95,98 \text{ Porc. sólidos} + 900,5 \text{ Vel. de giro}$$

Por otra parte, se obtuvo un MSE de 166.807, lo que se tradujo en un RMSE de 408,4 kW, es decir, un 6,57% respecto al promedio de la variable objetivo para este escenario (6.216,37 kW).

Finalmente, la distribución de los valores estimados por el modelo versus los valores reales para la potencia del molino arrojó un valor promedio de 0,493 para R^2 lo que se ve reflejado en la Figura 37.

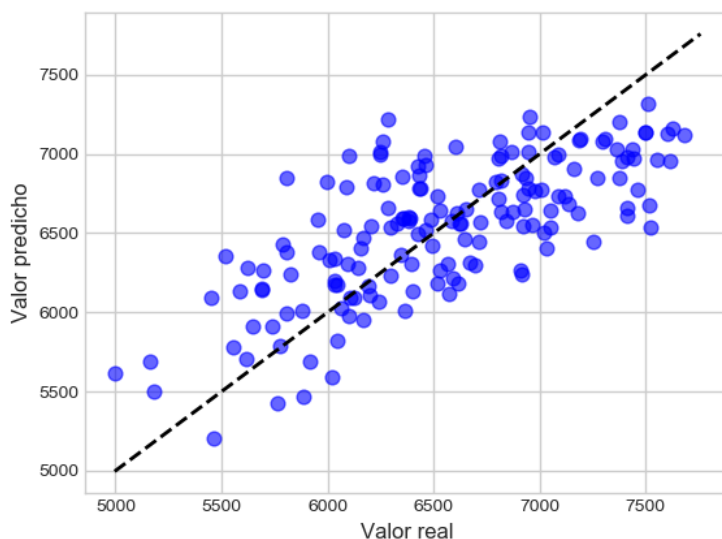


Figura 37: Distribución de valores reales vs predichos por RLM en escenario SAG 01, donde la línea punteada es la función identidad.

4.3.1.2 SAG 02

De la misma forma, pero utilizando los datos de la segunda línea de molienda, se obtuvo la siguiente ecuación del modelo:

$$\text{Potencia} = -5.092 + 83,99BWi - 0,02237 \text{ Tratamiento} - 2,813 P80 \\ + 77,36 \text{ Porc.sólidos} + 643,2 \text{ Vel.de giro}$$

Para este escenario, se obtuvo un MSE de 138.586, lo que se tradujo en un RMSE de 372,2 kW, es decir, un 5,86% respecto al promedio de la variable objetivo para este escenario (6.348,06 kW).

Por último, la distribución de los valores estimados por el modelo versus los valores reales para la potencia del molino entregó un valor promedio de 0,510 para R^2 lo que se ve reflejado en la Figura 38.

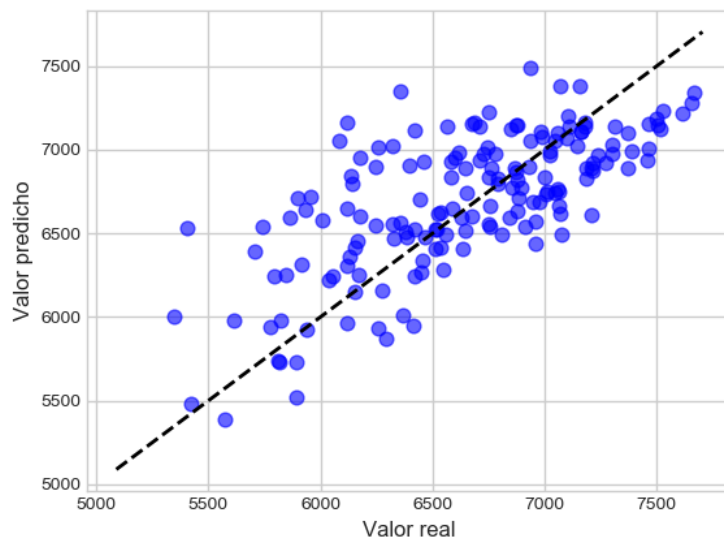


Figura 38: Distribución de valores reales vs predichos por RLM en escenario SAG 02, donde la línea punteada es la función identidad.

4.3.1.3 SAG 03

Por último, con los datos del escenario de la tercera línea de molienda, se obtuvo la siguiente ecuación del modelo:

$$\text{Potencia} = -1.085 + 461,8BWi - 0,02118 \text{ Tratamiento} - 6,373 P80 \\ - 31,77 \text{ Porc.sólidos} + 1.952 \text{ Vel.de giro}$$

Para este escenario, se obtuvo un MSE de 1.157.656, lo que permitió calcular un RMSE de 1.075,9 kW, es decir, un 5,35% respecto al promedio de la variable objetivo para este escenario (20.104,62 kW).

Por último, la distribución de los valores estimados por el modelo versus los valores reales para la potencia del molino entregó un valor promedio de 0,456 para R^2 lo que se ve reflejado en la Figura 39.

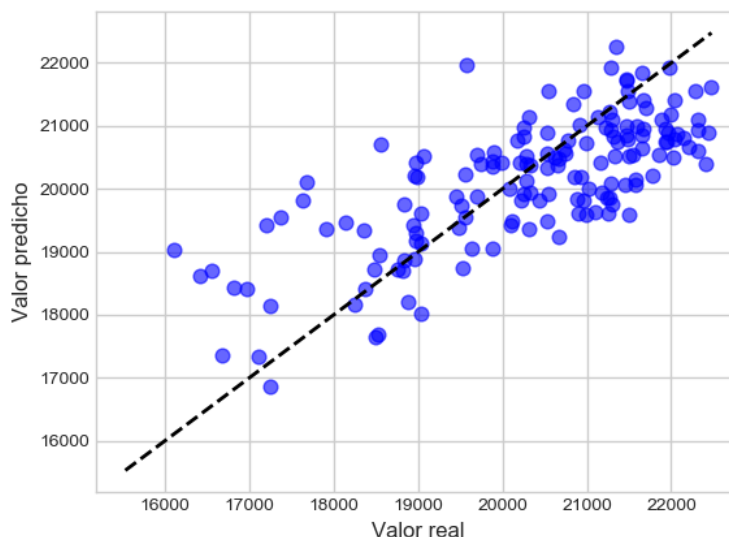


Figura 39: Distribución de valores reales vs predichos por RLM en escenario SAG 03, donde la línea punteada es la función identidad.

4.3.2 Método 2: K-Nearest neighbors (KNN)

Para este algoritmo la literatura sugiere recurrir a la normalización o estandarización de las bases de datos, esto debido a que, como se explica anteriormente (Ver sección 2.2.2), KNN trabaja con el cálculo de distancias euclidianas, lo que vuelve este tipo de modelos particularmente sensible a la diferencia de ordenes de magnitud entre una variable *input* y otra. Por lo anterior el analista de datos optó por la estandarización de datos mediante la técnica *z-score* antes de realizar el entrenamiento de los modelos.

Los tres modelos desarrollados mediante el método de *K-Nearest neighbors*, al igual que los anteriores (Regresión lineal múltiple) utilizaron el método de validación cruzada *Repeated K-fold* (Ver sección 2.3.2.3), manteniendo la cantidad de conjuntos en tres y realizando igualmente cien iteraciones aleatorias con distintos conjuntos generados. Posteriormente se calculó el error promedio de todas las iteraciones, mediante el cálculo del *Mean Squared*

Error (MSE) y *Root Mean Squared Error* (RMSE) (Ver sección 2.3.1). De igual forma se obtuvo el R^2 de los valores predichos en relación con los valores reales.

Para obtener mejores resultados se buscó el error mínimo mediante iteración ajustando el parámetro de “*n° neighbors*” que indica la cantidad de valores más cercanos que el algoritmo tomará en cuenta para estimar el valor de la variable objetivo, siendo 8 el número que mejores resultados entregó en los tres casos. Se debe considerar que un valor muy alto de este parámetro puede conducir a mejores resultados de RMSE, pero caer en el problema del *overfitting* (Ver sección 2.2.2).

4.3.2.1 SAG 01

Aplicando este algoritmo para los datos de la primera línea de molienda se obtuvo un MSE de 164.191, lo que permitió calcular un RMSE de 405,2 kW, es decir, un 6,51% respecto al promedio de la variable objetivo para este escenario (6.216,37 kW).

Por otra parte, la distribución de los valores predichos por el modelo KNN versus los valores reales para la potencia del molino arrojó un valor promedio de 0,502 para R^2 lo que se ve reflejado en la Figura 40.

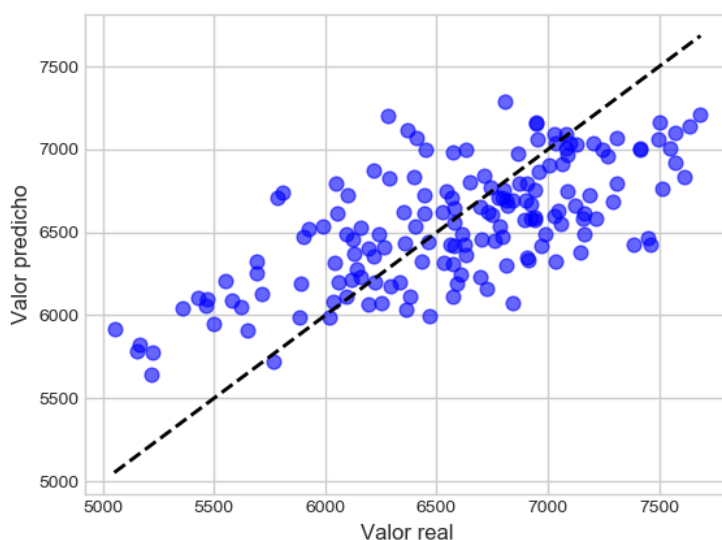


Figura 40: Distribución de valores reales vs predichos por KNN en escenario SAG 01, donde la línea punteada es la función identidad.

4.3.2.2 SAG 02

Para el segundo escenario los valores obtenidos fueron 143.013 para el MSE y, por lo tanto, un RMSE de 378,1 kW, equivalente a un 5,95% del valor promedio de la variable objetivo (6.348,06 kW).

Siguiendo el trabajo realizado en los otros casos, se generó el gráfico de dispersión de la potencia predicha versus la real entregada por el conjunto de datos de prueba. Obteniendo un valor de correlación R^2 de 0,495 y la Figura 41.

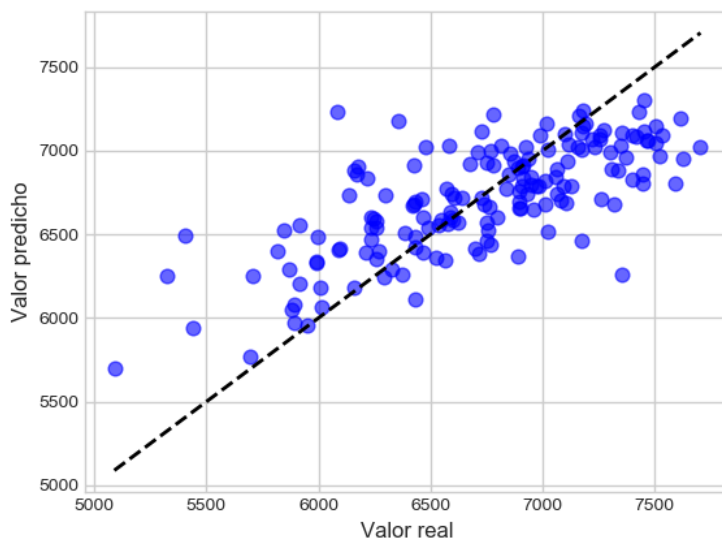


Figura 41: Distribución de valores reales vs predichos por KNN en escenario SAG 02, donde la línea punteada es la función identidad.

4.3.2.3 SAG 03

Por último, en el tercer escenario KNN entregó valores de 1.182.446 para el MSE, con el que se calcula un RMSE de 1.087,4 kW, equivalente a un 5,40% del promedio de la potencia promedio del equipo instalado en la línea número tres (20.104,62 kW).

Al graficar la dispersión de los valores reales versus los valores predichos se obtuvo un R^2 de 0,447 y la Figura 42.

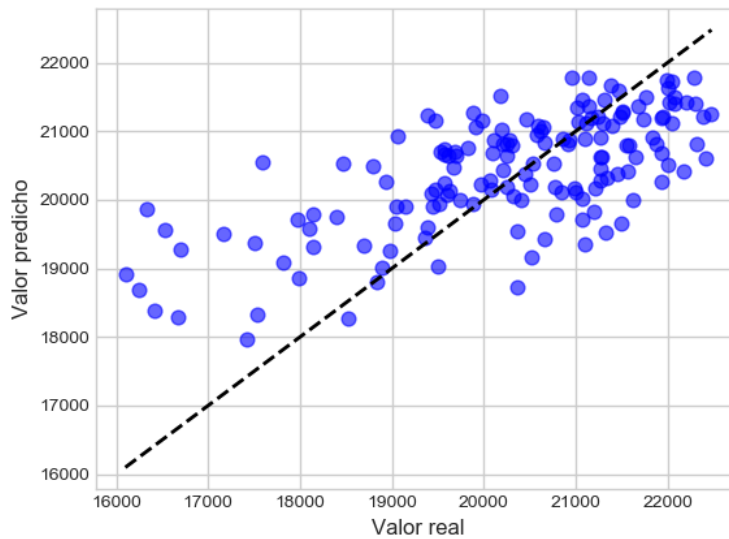


Figura 42: Distribución de valores reales vs predichos por KNN en escenario SAG 03, donde la línea punteada es la función identidad.

Para este escenario en particular se apreciaron observaciones mucho más dispersas a valores bajos de potencia del equipo, y, por lo tanto, residuos más altos. Lo anterior se puede explicar por la baja cantidad de datos que existen para esta línea de molienda operando en estos rangos. Se destaca entonces el mal funcionamiento del algoritmo KNN en estas condiciones, las cuales no deberían ocurrir con bases de datos ideales.

4.3.3 Método 3: Random forest

En la creación de modelos mediante el método Random Forest no es estrictamente necesaria la previa normalización o estandarización de los datos ya que este se basa en la construcción de árboles de decisión (Ver sección 2.2.4), los que son invariantes ante transformaciones monótonas de las características, es decir, transformaciones que mantienen el orden de los datos, por ejemplo, multiplicar por un valor positivo (desviación estándar) o sumar una constante (media). Sin embargo, se sugiere realizar pruebas empíricas en cada conjunto de datos para analizar cómo afecta el rendimiento de los modelos.

Para los tres modelos desarrollados con ayuda del algoritmo *Random Forest* se utilizó el método de validación cruzada *GridSearch* (Ver sección 2.3.2.4) para evaluar y validar sus rendimientos. Esta técnica realizó la búsqueda de los mejores hiperparámetros principales (Ver sección 2.2.4) teniendo como objetivo la optimización del *Mean Squared Error* (MSE) (Ver sección 2.3.1). Luego se entrenó el modelo con los mejores valores encontrados y se analizó la distribución de los valores estimados respecto a los valores reales, calculando adicionalmente el R^2 .

4.3.3.1 SAG 01

Para caso del primer escenario utilizando el algoritmo RF se definió un set de datos de prueba del 30% respecto al total. La validación cruzada *GridSearch* entregó los mejores hiperparámetros buscando el menor MSE, los cuales se muestran en la Tabla 13.

Tabla 13: Mejores hiperparámetros en modelo RF para SAG01. Estimados mediante GridSearch CV.

Hiperparámetros	
Max depth	14
Min samples leaf	1
Min samples split	13
N estimators	55

Se le asignó un valor de 5 a la cantidad de subconjuntos para la realización de la validación cruzada, obteniéndose un MSE de 151.605, lo que permitió el cálculo de un RSME de 389,3 kW, equivalente al 6,26% del promedio de la variable objetivo (6.216,37 kW).

Al analizar la dispersión de los valores predichos por el modelo y los valores reales de prueba se obtuvo un R^2 de 0,519 y la Figura 43.

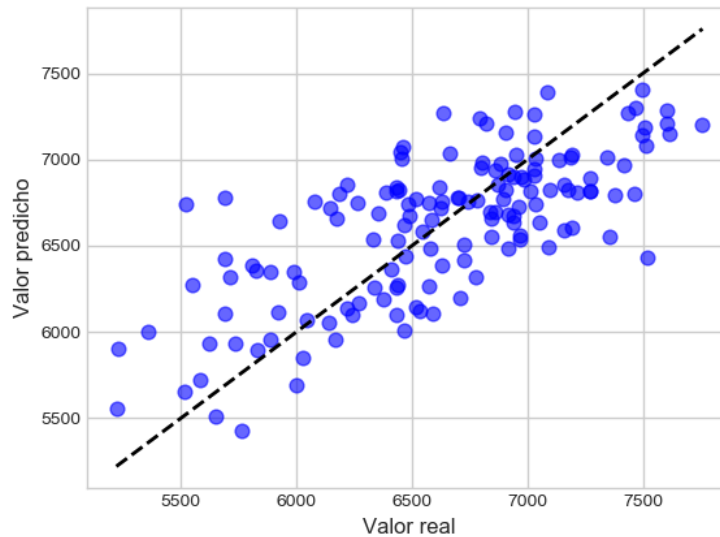


Figura 43: Distribución de valores reales vs predichos por RF en escenario SAG 01, donde la línea punteada es la función identidad.

4.3.3.2 SAG 02

En el segundo escenario se mantuvo el tamaño del conjunto de prueba del 30% de las observaciones totales. La técnica *GridSearch* arrojó los mejores hiperparámetros, los que se muestran en la Tabla 14.

Tabla 14: Mejores hiperparámetros en modelo RF para SAG02. Estimados mediante GridSearch CV.

Hiperparámetros	
Max depth	10
Min samples leaf	3
Min samples split	2
N estimators	76

Al igual que en el primer escenario se asignó un valor de 5 subconjuntos para la validación cruzada mediante *K-Fold*, alcanzando un resultado de 104.586 para el MSE y, por lo tanto, un RMSE de 323,3 kW, lo que equivale al 5,09% de la potencia media para esta base de datos (6.348,06 kW).

Por otra parte, al estudiar la distribución de los resultados predichos por RF versus los valores reales del set de prueba se calculó un R^2 de 0,598 y la Figura 44.

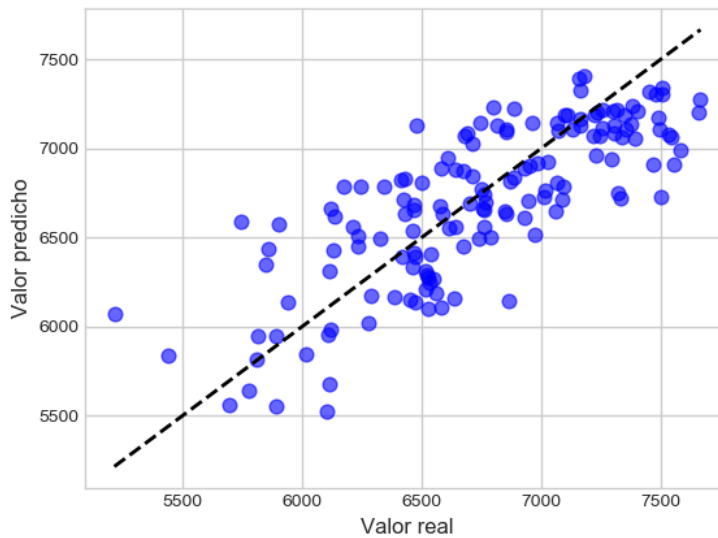


Figura 44: Distribución de valores reales vs predichos por RF en escenario SAG 02, donde la línea punteada es la función identidad.

4.3.3.3 SAG 03

Para la base de datos de la tercera línea de procesamiento se mantuvieron los parámetros generales de los escenarios anteriores, es decir, un tamaño de conjunto de prueba del 30% de los datos y 5 subconjuntos para la validación cruzada. Adicionalmente el *GridSearch* entregó los hiperparámetros óptimos, mostrados en la Tabla 15.

Tabla 15: Mejores hiperparámetros en modelo RF para SAG03. Estimados mediante GridSearch CV.

Hiperparámetros	
Max depth	13
Min samples leaf	3
Min samples split	2
N estimators	95

Utilizando los datos anteriores para el entrenamiento del modelo se obtuvo un MSE de 1.010.160, valor con el que se calculó un RMSE de 1.005,1 kW equivalente al 4,99% del valor promedio de potencia para el equipo de molienda instalado en la tercera línea (20.104,62 kW).

Por último, el R^2 obtenido para este caso es de 0,449 y al graficar la dispersión de los datos predichos versus los reales, se obtuvo lo mostrado en la Figura 45, donde nuevamente se pudo apreciar una menor precisión en los intervalos de menores valores para este escenario.

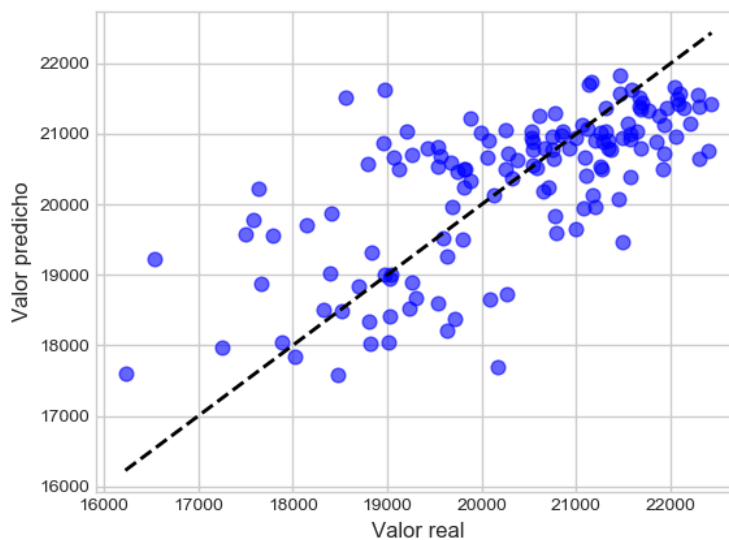


Figura 45: Distribución de valores reales vs predichos por RF en escenario SAG 03, donde la línea punteada es la función identidad.

4.3.4 Método 4: Gradient boosting

Para este estudio el analista de datos decidió utilizar el algoritmo *XGBoost* (Ver sección 2.2.5) debido a su alta eficiencia computacional. De esta manera se redujeron los tiempos de entrenamiento y búsqueda de hiperparámetros óptimos. Se descartó el uso de *LightGBM* ya

que la baja cantidad de datos disponibles no amerita aumentar la velocidad de entrenamiento en sacrificio de otras virtudes del primer algoritmo. Sin embargo, su aplicación también podría ser evaluada en otro estudio.

4.3.4.1 SAG 01

En el caso de los datos de la primera línea de molienda se utilizó un tamaño de muestra de test del 30% de las observaciones totales y 5 subconjuntos de datos para realizar la validación cruzada mediante el método *K-fold* (Ver sección 2.3.2.2). Los mejores hiperparámetros encontrados mediante el método *GridSearch* se muestran en la Tabla 16.

Tabla 16: Mejores hiperparámetros en modelo XGB para SAG01. Estimados mediante GridSearch CV.

Hiperparámetros	
Max depth	2
N estimators	120
Learning rate	0,1
Colsample bytree	0,8
Subsample	0,5

Al entrenar el modelo con los parámetros anteriores y evaluar sus métricas de error, se obtuvo un MSE de 157.074, lo que permitió calcular un RMSE de 396,3 kW, equivalente al 6,37% del promedio de la potencia en este periodo (6.216,37 kW).

Al generar el gráfico de dispersión de datos entre los valores de las variables predichas y las reales (Ver Figura 46) se obtuvo un R^2 de 0,502. Se pudieron observar residuos homogéneos a lo largo de todo el rango de las variables.

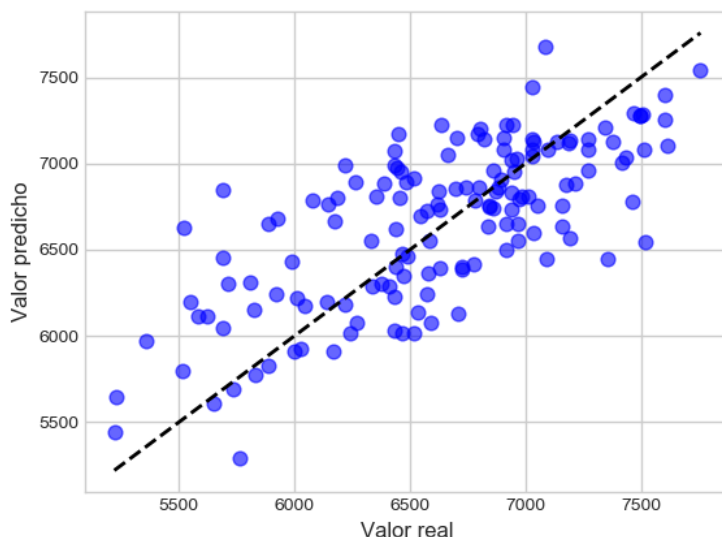


Figura 46: Distribución de valores reales vs predichos por XGB en escenario SAG 01, donde la línea punteada es la función identidad.

4.3.4.2 SAG 02

En el segundo escenario se mantuvieron los parámetros de tamaño de la muestra de prueba y cantidad de subconjuntos generados para realizar la validación cruzada de los hiperparámetros (30% y 5, respectivamente).

Los hiperparámetros encontrados por el algoritmo de *GridSearch* se enseñan en la Tabla 17.

Tabla 17: Mejores hiperparámetros en modelo XGB para SAG02. Estimados mediante GridSearch CV.

Hiperparámetros	
Max depth	3
N estimators	120
Learning rate	0,06
Colsample bytree	0,8
Subsample	0,43

Al entrenar el modelo final con los parámetros anteriores se obtuvo un MSE de 104.829, a partir del que se calcula el RMSE de 323,7 kW, que es igual a un 5,09% del promedio de los valores de la variable objetivo para este escenario (6.348,06 kW).

Al igual que en el caso anterior se dispusieron los datos estimados por el modelo y se graficaron en función de los datos reales de testeo. Obteniendo como resultado un R^2 de 0,597 y la Figura 47.

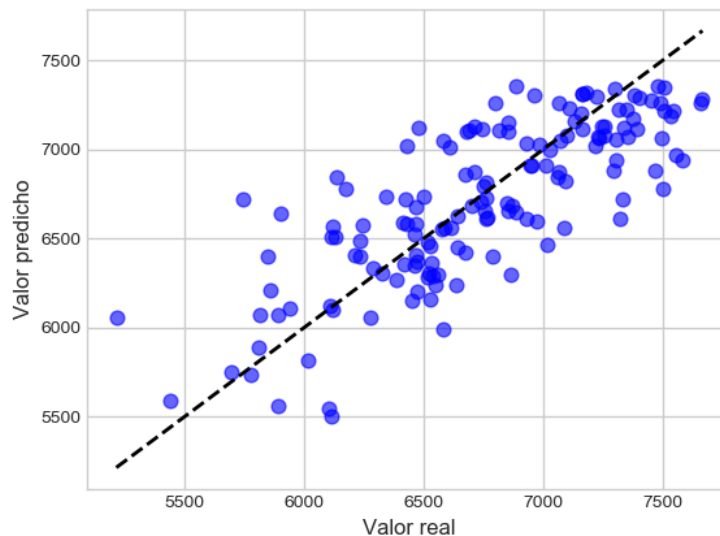


Figura 47: Distribución de valores reales vs predichos por XGB en escenario SAG 02, donde la línea punteada es la función identidad.

4.3.4.3 SAG 03

Para el último escenario se mantuvieron los parámetros utilizados en los dos anteriores: un tamaño de muestra de test del 30% y 5 subconjuntos para la validación mediante *K-fold*. De igual forma se buscaron los mejores hiperparámetros mediante *GridSearch*, mostrándolos en la Tabla 18.

Tabla 18: Mejores hiperparámetros en modelo XGB para SAG03. Estimados mediante *GridSearch CV*.

Hiperparámetros	
Max depth	4
N estimators	82
Learning rate	0,07
Colsample bytree	0,6
Subsample	0,5

El modelo XGB entrenado con los resultados anteriores presenta un MSE de 980.577, con lo que se calculó un RMSE de 990,2 kW. Este último valor representa un 4,92% de la potencia promedio del equipo estudiado (20.104,62 kW).

Por último, se realizó el mismo análisis de distribución de datos al representar los valores reales versus los predichos por el modelo (Ver Figura 48). El R^2 en este caso fue 0,465.

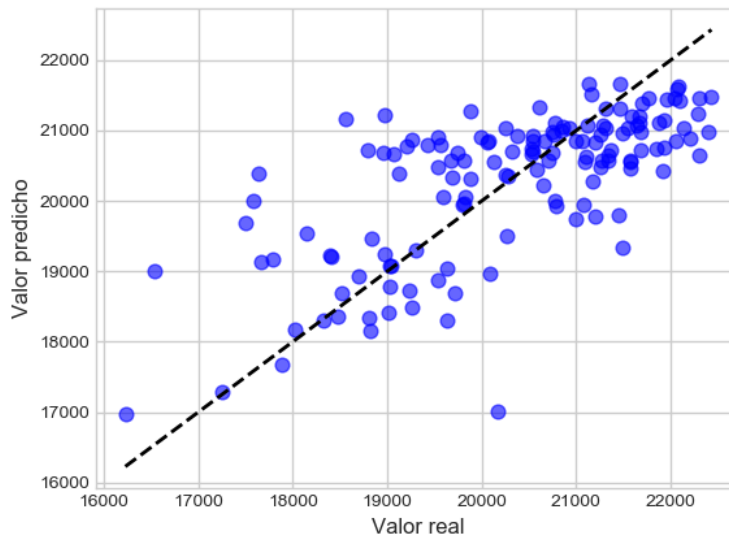


Figura 48: Distribución de valores reales vs predichos por XGB en escenario SAG 03, donde la línea punteada es la función identidad.

La zona de menores potencias enseñada en la Figura 48 (16.000-19.000 kW) muestra altos valores residuales y baja precisión. Lo anterior puede deberse a la baja cantidad de observaciones en esta zona, lo que induce un pobre entrenamiento del modelo y perjudica los resultados de la zona más alta y precisa.

4.4 COMPARACIÓN DE MÉTODOS

Para llevar a cabo esta parte del estudio, el primer paso fue comparar los valores de R^2 para cada escenario, por lo que se generó la Tabla 19. Preliminarmente, se pudo observar que las predicciones más precisas para el escenario 1 y 2 se obtuvieron mediante el modelo *Random Forest*. En el caso de la línea de molienda 3, el modelo que mejor se adaptó fue el de *XGBoost*.

Tabla 19: Valores de correlación entre potencia predicha y real para todos los escenarios.

	SAG 01	SAG 02	SAG 03
RLM	0,493	0,510	0,456
KNN	0,502	0,495	0,447
RF	0,519	0,598	0,449
XGB	0,502	0,597	0,465

De forma paralela se generó la Tabla 20, la que muestra los errores relativos porcentuales calculados al hacer el cociente entre los respectivos RMSE y la media de los valores reales de potencia para cada equipo SAG. Esta tabla presentó datos acordes a la Tabla 19 ya que indicó que los menores errores se obtuvieron utilizando los mismos algoritmos previamente señalados para cada caso. La única diferencia es que para este caso *Random Forest* y *XGBoost* tuvieron un desempeño aproximadamente igual para el caso del SAG 02, resultado esperable al considerar que las correlaciones solo difieren en una milésima.

Tabla 20: Valores de error porcentual en la estimación de la potencia, calculado a partir del RMSE en cada escenario.

	SAG 01	SAG 02	SAG 03
RLM	6,57%	5,86%	5,35%
KNN	6,51%	5,95%	5,40%
RF	6,26%	5,09%	4,99%
XGB	6,37%	5,09%	4,92%

Con los anteriores resultados, se confirma que los algoritmos más complejos (RF y XGB) entregan mejores predicciones que los de más simple funcionamiento (RLM y KNN). Sin embargo, la diferencia de los errores se mantiene en el orden de las décimas y hay que considerar que los modelos más complejos son preferentemente sensibles al *overfitting*, por lo que al utilizarlos en un conjunto de datos distinto al estudiado podrían no presentar tan buenos resultados como en este caso, ya sean datos de otro periodo en la misma faena (MLP) o de una faena distinta.

Por otra parte, se observó que la tercera línea de molienda tiene los peores valores de correlación entre valores estimados y reales, esto a pesar de tener los errores porcentuales más bajos entre los datos de los tres equipos SAG.

Analizando los gráficos de dispersión para el tercer escenario (Figuras 39, 42, 45 y 48) se pudo observar que en todos los casos la zona de bajos valores de potencia del molino presentó

una baja precisión en las predicciones. Como ya se mencionó anteriormente, esto puede explicarse por la baja densidad de observaciones en esta zona, a causa de esto ningún algoritmo fue capaz de entrenarse bien aquí. Este efecto puede minimizarse al aumentar la frecuencia de muestreo, es decir, obtener potencias promedio cada intervalos más cortos de tiempo (1 hora, 4 horas, etc.). Lo anterior es un indicador de que, si solo se dispone de un valor de potencia promedio al día, no se podrán realizar predicciones confiables para la toma de decisiones operacionales, independiente de la precisión y calidad del modelo de aprendizaje automático utilizado.

Con el fin de observar el comportamiento de las predicciones a través de una parte del periodo de tiempo estudiado, se crearon gráficos de línea para los modelos con mejor rendimiento para cada una de las líneas de procesamiento, es decir, los resultados de *Random Forest* para SAG 01 y SAG 02, mientras que para el SAG 03 se considerarán los resultados de *XGBoost*. En los tres casos se utilizaron los parámetros obtenidos anteriormente, pero se reentrena con el primer 70% de las observaciones (ordenadas cronológicamente) y se evalúa con el último 30%, equivalente aproximadamente a un periodo de 6 meses de operación.

Para el escenario SAG 01 se obtuvo la Figura 49 (Ver Figuras 55 y 56 en Anexos sección a) para más detalle). Mientras que para el escenario SAG 02 se generó la Figura 50 (Ver Figuras 57 y 58 en Anexos sección a) para más detalle). Por último, para el caso del SAG 03 se obtuvo la Figura 51 (Ver Figuras 59 y 60 en Anexos sección a) para más detalle).

De manera general se pudo observar que para los tres casos las predicciones siguen de forma bastante cercana la tendencia de los datos reales. Sin embargo, para los escenarios 1 y 2 se pudo apreciar cierta incapacidad de los modelos para seguir las observaciones de mayor desviación, esto podría deberse nuevamente a la baja densidad de datos o a que no se está considerando como *input* alguna variable capaz de explicar esta parte de la varianza.

Por otra parte, en el caso del SAG 03 se observa que los valores estimados en general siguen mejor la tendencia. Esto podría deberse a que este escenario muestra una operación del equipo más regular en el tiempo y puede resultar más sencillo para el modelo explicar la varianza.

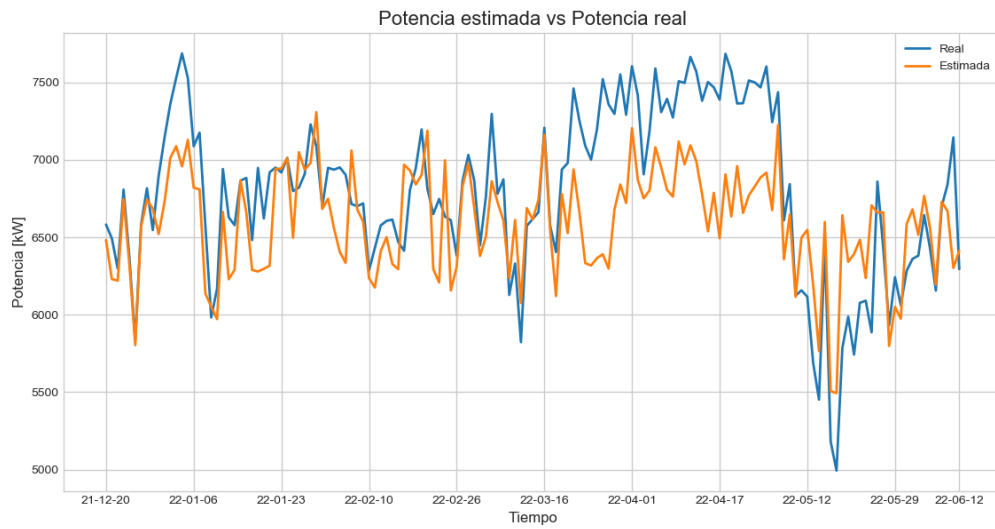


Figura 49: Gráfico de potencia real vs predicha utilizando modelo RF para SAG 01 en un periodo de 6 meses.

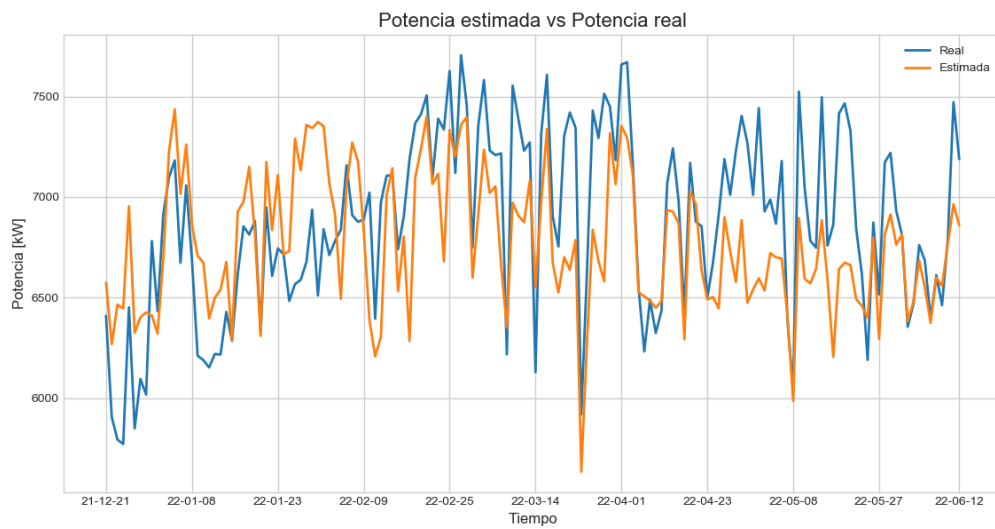


Figura 50: Gráfico de potencia real vs predicha utilizando modelo RF para SAG 02 en un periodo de 6 meses.

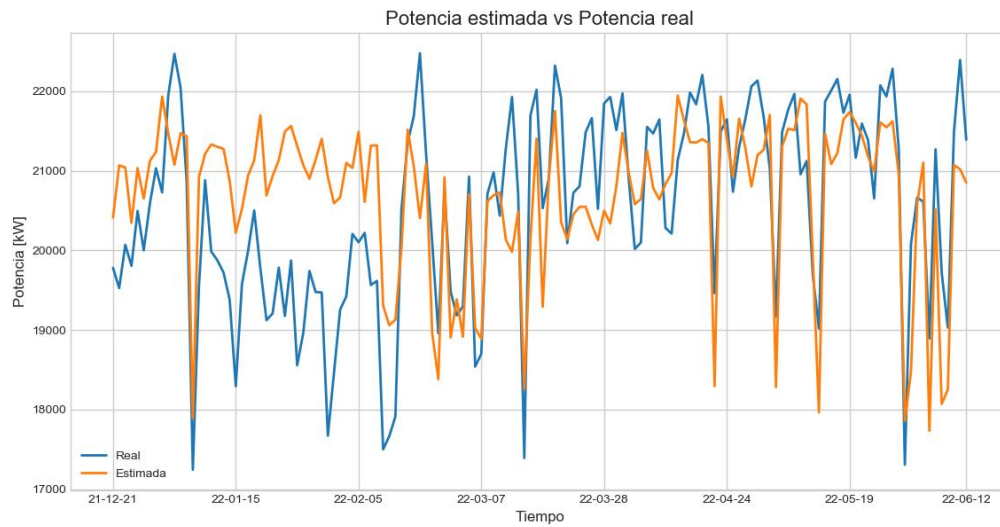


Figura 51: Gráfico de potencia real vs predicha utilizando modelo XGB para SAG 03 en un periodo de 6 meses.

Adicionalmente, se observó que en ciertos *peaks* las curvas de datos predichos seguían demasiado la tendencia de las curvas reales, mostrando un posible fenómeno de *overfitting* a causa de la gran correlación existente entre potencia del molino y su predictor “velocidad de giro”. Considerando además que la información que entrega este predictor no se encontraría disponible previamente en una posible implementación en tiempo real, se optó por estudiar brevemente y de manera cualitativa el comportamiento de los mismos tres modelos finales, pero siendo entrenados y validados sin considerar la velocidad de giro como una variable *input*. Así se obtuvieron los resultados mostrados en las Figuras 52, 53 y 54.

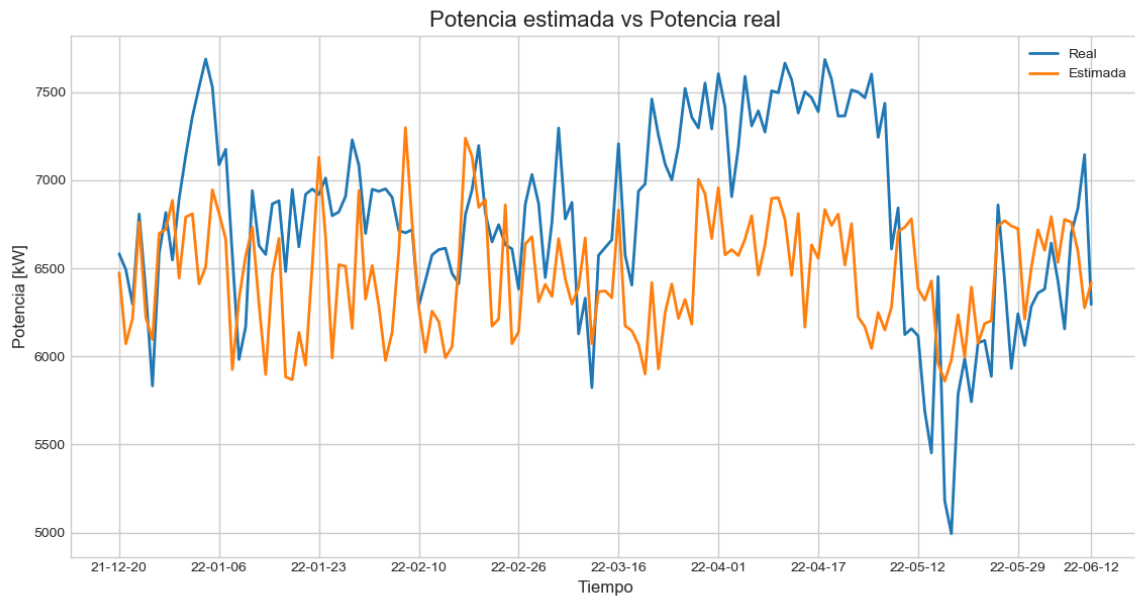


Figura 52: Gráfico de potencia real vs predicha utilizando modelo RF para SAG 01 sin considerar velocidad de giro como input.

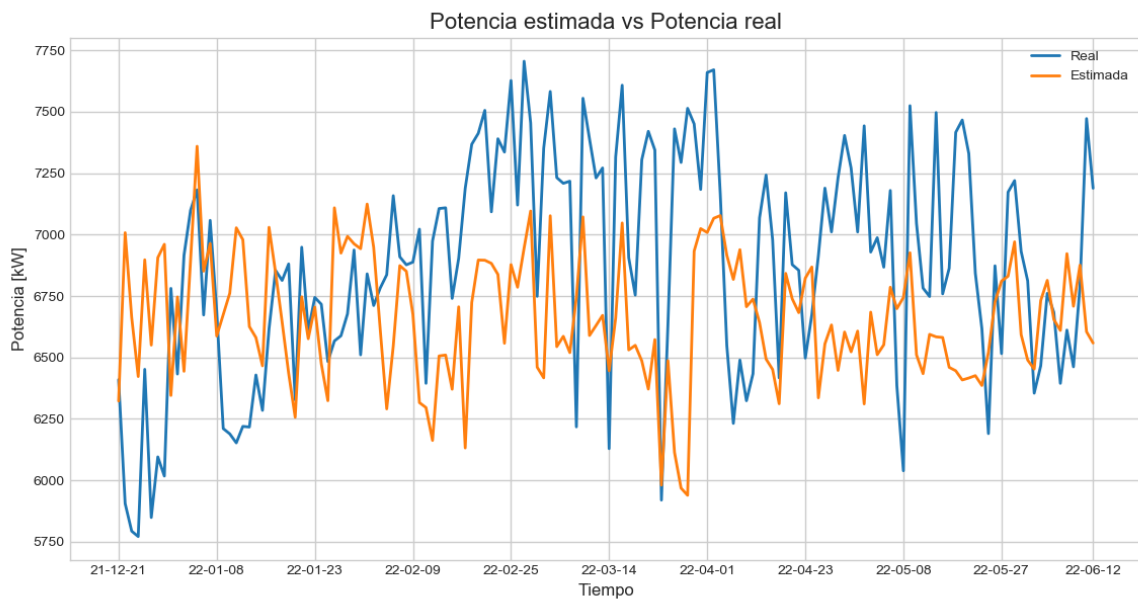


Figura 53: Gráfico de potencia real vs predicha utilizando modelo RF para SAG 02 sin considerar velocidad de giro como input.

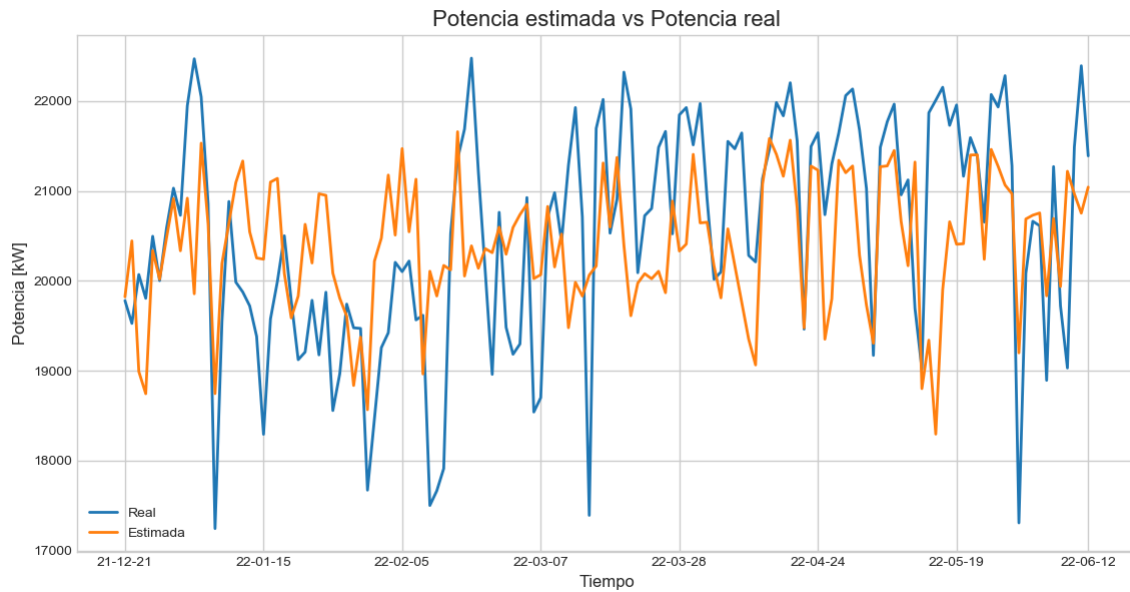


Figura 54: Gráfico de potencia real vs predicha utilizando modelo XGB para SAG 03 sin considerar velocidad de giro como input.

Se pudo observar que el rendimiento de los modelos baja considerablemente al comparar con los generados considerando velocidad de giro (Ver Figuras 49, 50 y 51), llegando a existir periodos donde la tendencia de los datos predichos va completamente en dirección opuesta a la de los datos reales. En particular, el segundo escenario es el que peor se adapta. A pesar de lo anterior, también se aprecian zonas de buen rendimiento en los tres distintos casos. Se observó también que las curvas de valores estimados son mucho más suavizadas (menor variabilidad) que en los casos anteriores.

CAPÍTULO V: CONCLUSIONES

La presente memoria de título logró desarrollar de la manera esperada todos los modelos planteados para los tres distintos escenarios y utilizando valores históricos de las distintas variables mencionadas se lograron obtener resultados de precisión bastante satisfactorios considerando la baja densidad de datos disponibles. Adicionalmente, se estudió y utilizó el estado del arte para seleccionar y considerar solamente información que pudiera ser provechosa para el desarrollo de los ya mencionados modelos.

Los modelos fueron desarrollados bajo la simplificación de que no existen más variables independientes que influyan en el consumo de potencia eléctrica en los equipos que las utilizadas como *inputs*. También se toma como suposición que el comportamiento de los tres equipos se moverá dentro de los mismos rangos que muestra en las bases de datos dentro de cualquier periodo operacional. La creación de estos modelos estadísticos fue satisfactoriamente realizada con algoritmos de aprendizaje automático (o *machine learning*) de modo que al replicar la metodología de esta memoria los resultados y los hiperparámetros calibrados pueden variar ligeramente, ya que estos dependen directamente de la aleatoriedad de la selección de conjuntos de entrenamiento.

Para los dos primeros escenarios (SAG01 y SAG02) los mejores resultados se obtuvieron mediante el uso del algoritmo *Random forest*. Para la línea SAG01 los hiperparámetros calibrados fueron: *max depth* = 14, *min samples leaf* = 1, *min samples split* = 13, *n estimators* = 55, con lo que se obtuvo un RMSE de 6,27% y un valor de correlación entre las potencias predichas y reales (R^2) de 0,519. En el segundo caso, es decir, SAG02, los hiperparámetros calibrados fueron: *max depth* = 10, *min samples leaf* = 3, *min samples split* = 2, *n estimators* = 76, llegando a un RMSE de 5.09% y un R^2 de 0,598. Por último, el escenario del SAG03 mostró resultados más precisos al trabajar con el algoritmo *Gradient boosting (XGBoost)* al calibrar los siguientes hiperparámetros: *max depth* = 4, *n estimators* = 82, *learning rate* = 0,07, *colsample bytree* = 0,6, *subsample* = 0,5. Mediante el modelo anterior se logró un RMSE de 4,92% y un índice de correlación R^2 de 0,465.

Por otra parte, el análisis indica que modelos en esta etapa de desarrollo tan temprano, no son suficiente para cumplir con las expectativas de una eventual implementación industrial en tiempo real, es decir, un algoritmo que sea capaz de entregar una estimación confiable de potencia requerida del equipo según las características del mineral que se planea alimentar a corto plazo.

La aceptable precisión de todos los modelos desarrollados depende en gran medida de la variable “velocidad de giro”, por lo que al no disponer de ella como predictor se perdería en gran porcentaje la confiabilidad de las estimaciones. Sin embargo, los dos algoritmos más complejos que se estudiaron (*Random forest* y *XGBoost*) se muestran prometedores para la futura creación de nuevos modelos entrenados con bases de datos más amplias.

Por otra parte, se observó que los rangos acotados de información, específicamente, la poca dispersión de valores en las variables puede conducir a los algoritmos a ser entrenados

solamente en condiciones cercanas al promedio, por lo que al momento de intentar predecir valores más lejanos a la media podría mostrar errores mucho mayores.

Se espera que el uso de metodologías de aprendizaje automático se expanda cada vez más en la industria, por lo que se insta a seguir fomentando este camino de estudio, sobre todo en usos que podrían tener un gran impacto económico, como puede ser la predicción del rendimiento y funcionamiento de grandes equipos en minería. Con este fin, se enlistan las posibles expansiones que podría tener esta investigación para ser más fructífera en posibles usos en terreno.

TRABAJO FUTURO PROPUESTO

De acuerdo con los resultados de este estudio, se propone realizar el siguiente trabajo a futuro con el fin de mejorar las predicciones de potencia en molinos SAG:

- Investigar algoritmos que no hayan sido abarcados en esta memoria, con el fin de tener una comparativa más amplia.
- Repetir el trabajo con bases de datos disponibles de otras faenas, con distintas mineralogías o distintos equipos. De esta forma sería posible probar el funcionamiento de los modelos en condiciones distintas, o en su defecto, crear modelos más generales basándose en la información de las distintas faenas en conjunto.
- Utilizar una mayor densidad de datos, en lo posible, con una frecuencia mayor a una hora, ya que la utilizada para este estudio (una observación al día) es insuficiente para anticipar decisiones operativas en caso de que se utilizara un modelo conectado en línea a un sistema de control en la industria.
- Evaluar una posible mejora de los rendimientos de los modelos creados con los algoritmos *Random Forest* y *Gradient Boosting* al realizar una previa estandarización de las bases de datos.
- Realizar un trabajo similar, pero cambiando la variable objetivo a predecir, por ejemplo, consumo de medios de molienda o desgaste de revestimientos. Esto, para estudiar el efecto que pueden tener los mismos regresores en otros aspectos que, al igual que la potencia, significan gran parte de los costes asociados al proceso de conminución.

REFERENCIAS

- Amat Rodrigo, J. (2021). Machine learning con Python y Scikit-learn. https://www.cienciadedatos.net/documentos/py06_machine_learning_python_scikitlearn.html
- Both, C., & Dimitrakopoulos, R. (2021). Applied Machine Learning for Geometallurgical Throughput Prediction—A Case Study Using Production Data at the Tropicana Gold Mining Complex. *Minerals*, 11(11), 1257.
- Laubscher, D. H. (1990). A geomechanics classification system for the rating of rock mass in mine design. *Journal of the Southern African Institute of Mining and Metallurgy*, 90(10), 257-273.
- Morell Muñoz, S. F. (2014). Propuesta de modificación de circuito para planta de flotación colectiva Cu-Mo en planta concentradora El Chacay, minera Los Pelambres.
- Olmos, B. (2018). Ensayos de conminución para evaluar el efecto de la aplicación de microondas a muestras minerales.
- Pasten Rivera, R. J. (2018). Homologación de prácticas operacionales y estandarización de procesos en planta concentradora Los Pelambres.
- Portal Minero. (2006). Manual General de minería y metalurgia.
- Rojo, J. M., & Abuín, J. M. R. (2007). Regresión lineal múltiple.
- Saez Catrileo, M. E. (2009). Caracterización estructural y geotécnica de los niveles superiores de la mina este del yacimiento los pelambres.
- Silva Balocchi, M. A. (2013). Variables operacionales y su influencia en la operación, pro optimización del consumo energético en la molienda SAG.
- Verret, F. O., Chiasson, G., & Mcken, A. (2011). SAG mill testing – an overview of the test procedures available to characterize ore grindability.
- Wambeke, T., Elder, D., Miller, A., Benndorf, J., & Peattie, R. (2018). Real-time reconciliation of a geometallurgical model based on ball mill performance measurements—a pilot study at the Tropicana gold mine. *Mining Technology*, 127(3), 115-130.

Anexos

a) Figuras

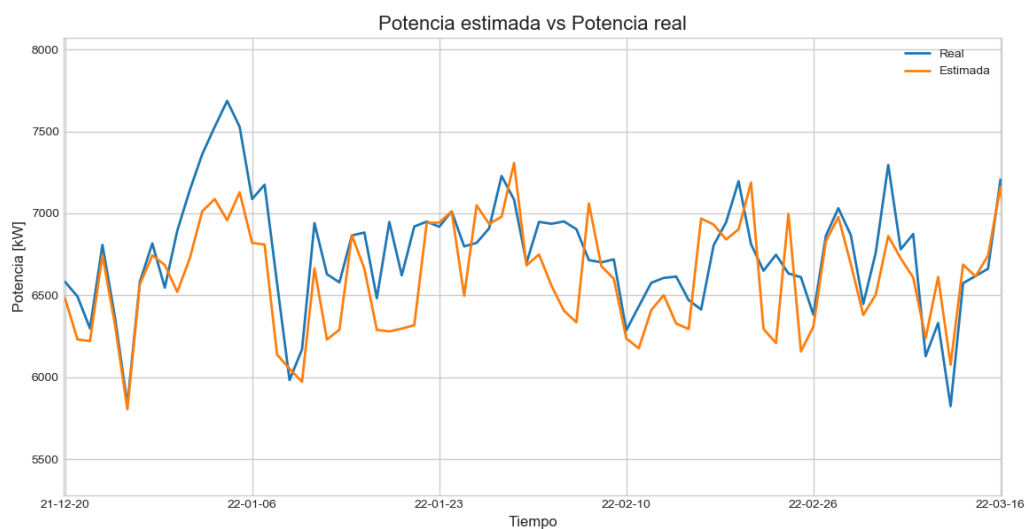


Figura 55: Primera sección de la Figura 49: Gráfico de potencia real vs predicha utilizando modelo RF para SAG 01 en un periodo de 6 meses.

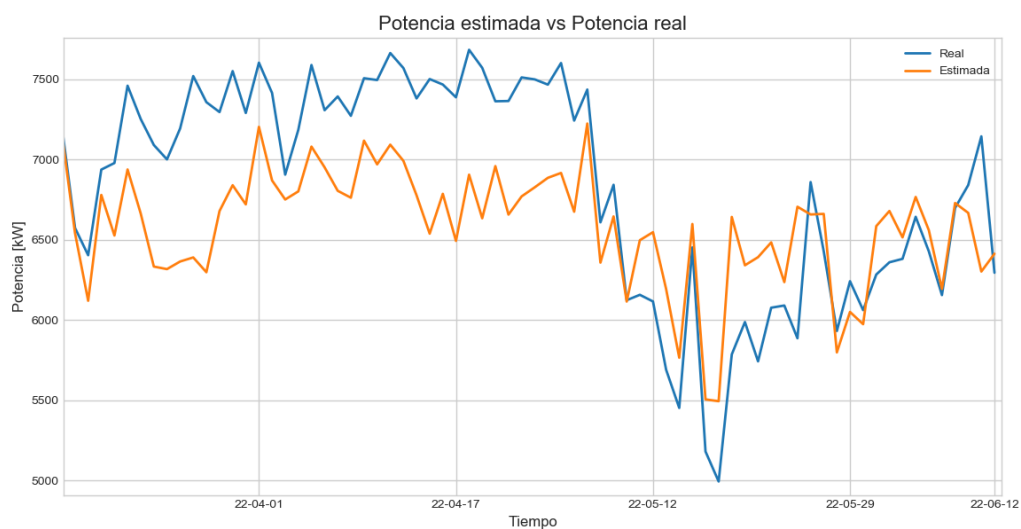


Figura 56: Segunda sección de la Figura 49: Gráfico de potencia real vs predicha utilizando modelo RF para SAG 01 en un periodo de 6 meses.

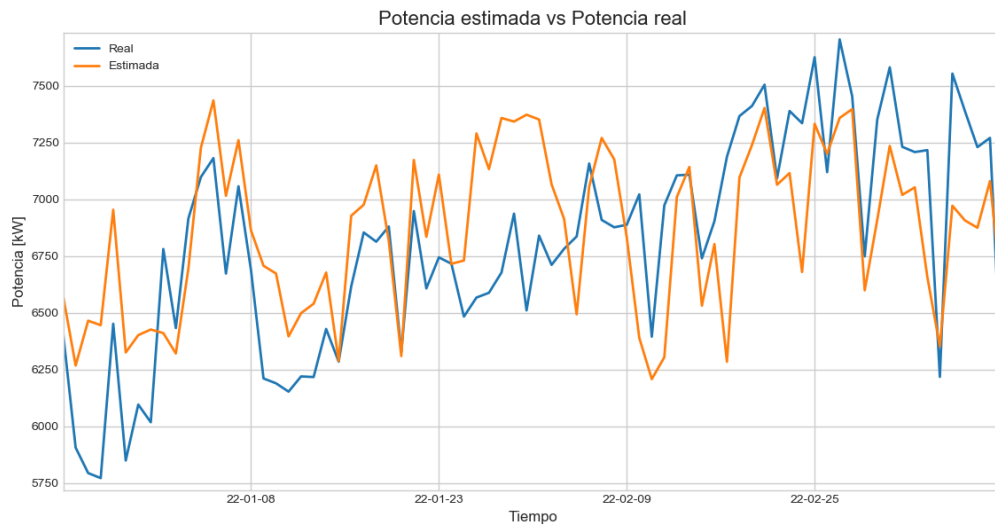


Figura 57: Primera sección de la Figura 50: Gráfico de potencia real vs predicha utilizando modelo RF para SAG 02 en un periodo de 6 meses.

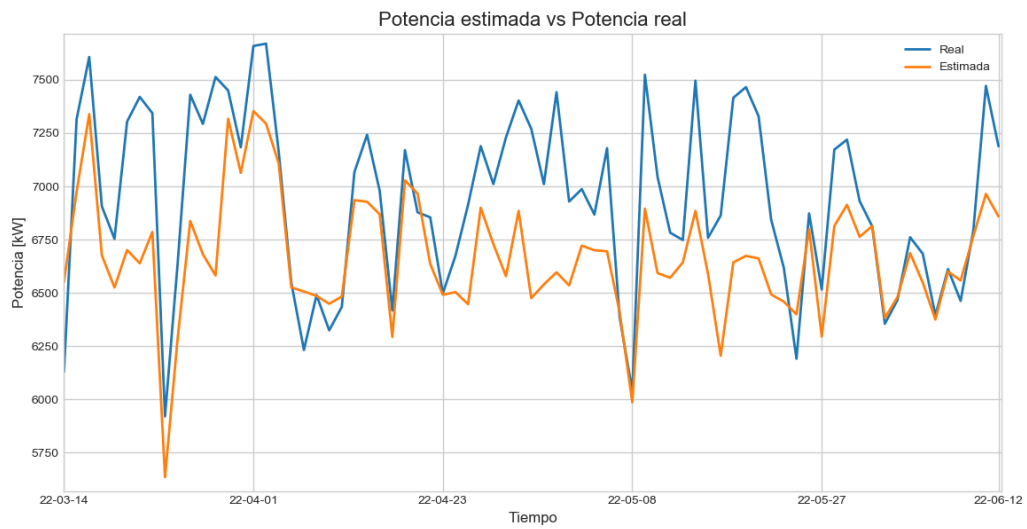


Figura 58: Segunda sección de la Figura 50: Gráfico de potencia real vs predicha utilizando modelo RF para SAG 02 en un periodo de 6 meses.

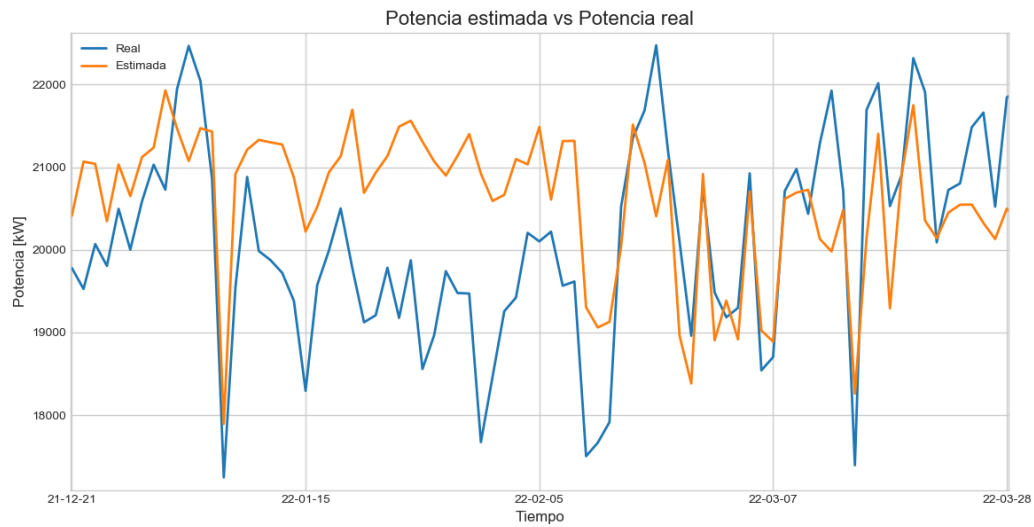


Figura 59: Primera sección de la Figura 51: Gráfico de potencia real vs predicha utilizando modelo XGB para SAG 03 en un periodo de 6 meses.

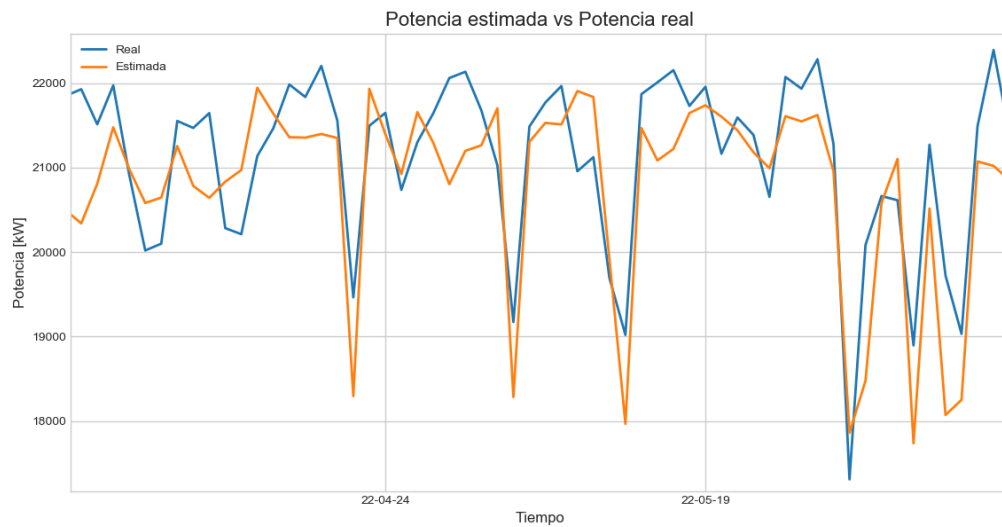


Figura 60: Segunda sección de la Figura 51: Gráfico de potencia real vs predicha utilizando modelo XGB para SAG 03 en un periodo de 6 meses.

b) Código Python

A modo de ejemplo y para una eventual repetición del estudio siguiendo la sección de trabajo futuro propuesto se adjunta el código utilizado para la limpieza de datos y desarrollo de modelos para el primer escenario.

```
import pandas as pd
import datetime
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```
DB1 = pd.read_excel("BD SAG 1.xlsx", header = 2).tail(-1)
DB1 = pd.DataFrame(DB1)
DB1.index = DB1["Fecha"]
DB1.index = pd.to_datetime(DB1.index, errors = "coerce").strftime("%Y-%m-%d")
DB1.drop(labels = "Fecha", axis = 1, inplace=True)
```

```
DB1["% sólidos L1"] = DB1["% sólidos L1"].astype('float64')
DB1["Veloc. Giro SAG01 (rpm)"] = DB1["Veloc. Giro SAG01 (rpm)"].astype('float64')
DB1["Potencia SAG01"] = DB1["Potencia SAG01"].astype('float64')
```

#Histogramas sin limpieza

```
fig, axs = plt.subplots(3, 3)
```

```
axs[0, 0].hist(DB1['BWi'], bins = 20, edgecolor = "black", facecolor = "aquamarine")
axs[0, 0].set_title('BWi', fontsize = 10)
```

```
axs[0, 1].hist(DB1['DWi'], bins = 20, edgecolor = "black", facecolor = "aquamarine")
axs[0, 1].set_title('DWi', fontsize = 10)
```

```
axs[0, 2].hist(DB1['Abx'], bins = 20, edgecolor = "black", facecolor = "aquamarine")
axs[0, 2].set_title('Axb', fontsize = 10)
```

```
axs[1, 0].hist(DB1['Tratamiento L1 (ton)'], bins = 20, edgecolor = "black", facecolor = "aquamarine")
axs[1, 0].set_title('Tratamiento L1 (ton)', fontsize = 10)
```

```
axs[1, 1].hist(DB1['P80 L1'], bins = 20, edgecolor = "black", facecolor = "aquamarine")
axs[1, 1].set_title('P80 L1', fontsize = 10)
```

```
axs[1, 2].hist(DB1['% sólidos L1'], bins = 20, edgecolor = "black", facecolor = "aquamarine")
axs[1, 2].set_title('% sólidos L1', fontsize = 10)
```

```
axs[2, 0].hist(DB1['Veloc. Giro SAG01 (rpm)'], bins = 20, edgecolor = "black", facecolor = "aquamarine")
axs[2, 0].set_title('Veloc. Giro SAG01 (rpm)', fontsize = 10)
```

```
axs[2, 1].hist(DB1['Horas de operación L1'], bins = 20, edgecolor = "black", facecolor = "aquamarine")
axs[2, 1].set_title('Horas de operación L1', fontsize = 10)
```

```
axs[2, 2].hist(DB1['Potencia SAG01'], bins = 20, edgecolor = "black", facecolor = "aquamarine")
axs[2, 2].set_title('Potencia SAG01', fontsize = 10)
```

```
fig.tight_layout()
plt.show()
```

#Limpieza de datos

```
DB1 = DB1.drop_duplicates()
DB1 = DB1.dropna()
DB1 = DB1.apply(pd.to_numeric, errors = "coerce")
```

```
# Limpiar tratamiento, horas operacion y velocidad de giro
DB1['Tratamiento L1 (ton)'].loc[(DB1['Tratamiento L1 (ton)'] <
np.nanpercentile(DB1['Tratamiento L1 (ton)'],6))] = np.nan
DB1['Veloc. Giro SAG01 (rpm)'].loc[(DB1['Veloc. Giro SAG01 (rpm)'] <
np.nanpercentile(DB1['Veloc. Giro SAG01 (rpm)'],7))] = np.nan
DB1['Horas de operación L1'].loc[(DB1['Horas de operación L1'] < np.nanpercentile(DB1['Horas de
operación L1'],9))] = np.nan
```

```
# Limpiar resto de variables por percentiles inferiores y superiores
```

```
DB1['BWi '].loc[(DB1['BWi '] < np.nanpercentile(DB1['BWi '],1))] = np.nan  
DB1['BWi '].loc[(DB1['BWi '] > np.nanpercentile(DB1['BWi '],99))] = np.nan
```

```
DB1['DWi'].loc[(DB1['DWi'] < np.nanpercentile(DB1['DWi'],1))] = np.nan  
DB1['DWi'].loc[(DB1['DWi'] > np.nanpercentile(DB1['DWi'],99))] = np.nan
```

```
DB1['Abx '].loc[(DB1['Abx '] < np.nanpercentile(DB1['Abx '],1))] = np.nan  
DB1['Abx '].loc[(DB1['Abx '] > np.nanpercentile(DB1['Abx '],99))] = np.nan
```

```
DB1['P80 L1'].loc[(DB1['P80 L1'] < np.nanpercentile(DB1['P80 L1'],3))] = np.nan  
DB1['P80 L1'].loc[(DB1['P80 L1'] > np.nanpercentile(DB1['P80 L1'],99))] = np.nan
```

```
DB1['% sólidos L1'].loc[(DB1['% sólidos L1'] < np.nanpercentile(DB1['% sólidos L1'],1))] =  
np.nan  
DB1['% sólidos L1'].loc[(DB1['% sólidos L1'] > np.nanpercentile(DB1['% sólidos L1'],98))] =  
np.nan
```

```
DB1['Potencia SAG01'].loc[(DB1['Potencia SAG01'] < np.nanpercentile(DB1['Potencia SAG01'],3))]  
= np.nan  
DB1['Potencia SAG01'].loc[(DB1['Potencia SAG01'] > np.nanpercentile(DB1['Potencia  
SAG01'],99))] = np.nan
```

```
DB1 = DB1.dropna()
```

```
# Histogramas tras limpieza
```

```
fig, axs = plt.subplots(3, 3)
```

```
axs[0, 0].hist(DB1['BWi '], bins = 20, edgecolor = "black", facecolor = "orange")  
axs[0, 0].set_title('BWi', fontsize = 10)
```

```
axs[0, 1].hist(DB1['DWi'], bins = 20, edgecolor = "black", facecolor = "orange")  
axs[0, 1].set_title('DWi', fontsize = 10)
```

```
axs[0, 2].hist(DB1['Abx '], bins = 20, edgecolor = "black", facecolor = "orange")  
axs[0, 2].set_title('Axb', fontsize = 10)
```

```
axs[1, 0].hist(DB1['Tratamiento L1 (ton)'], bins = 20, edgecolor = "black", facecolor =  
"orange")  
axs[1, 0].set_title('Tratamiento L1 (ton)', fontsize = 10)
```

```
axs[1, 1].hist(DB1['P80 L1'], bins = 20, edgecolor = "black", facecolor = "orange")  
axs[1, 1].set_title('P80 L1', fontsize = 10)
```

```
axs[1, 2].hist(DB1['% sólidos L1'], bins = 20, edgecolor = "black", facecolor = "orange")  
axs[1, 2].set_title('% sólidos L1', fontsize = 10)
```

```
axs[2, 0].hist(DB1['Veloc. Giro SAG01 (rpm)'], bins = 20, edgecolor = "black", facecolor =  
"orange")  
axs[2, 0].set_title('Veloc. Giro SAG01 (rpm)', fontsize = 10)
```

```
axs[2, 1].hist(DB1['Horas de operación L1'], bins = 20, edgecolor = "black", facecolor =  
"orange")  
axs[2, 1].set_title('Horas de operación L1', fontsize = 10)
```

```
axs[2, 2].hist(DB1['Potencia SAG01'], bins = 20, edgecolor = "black", facecolor = "orange")  
axs[2, 2].set_title('Potencia SAG01', fontsize = 10)
```

```
fig.tight_layout()  
plt.show()
```

```
# Análisis Multivariable
```

```
# Mapa de calor
```

```
corrmat = DB1.corr()
```

```
f, ax = plt.subplots(figsize=(8, 6))
sns.heatmap(corrmat, vmax=.8, square=True, cmap="RdYlGn", alpha = .8, annot = True, cbar =
True)
fig.tight_layout()

# Análisis bivariable, gráficos de dispersión.
#Relación entre durezas

plt.scatter(x = DB1["BWi "], y =DB1["DWi"], alpha = 0.6, c = "b", marker = "o", s= 60)
plt.xlabel("BWi")
plt.ylabel("DWi")

plt.scatter(x = DB1["BWi "], y =DB1["Abx "], alpha = 0.6, c = "b", marker = "o", s= 60)
plt.xlabel("BWi")
plt.ylabel("Axb")

plt.scatter(x = DB1["Abx "], y =DB1["DWi"], alpha = 0.6, c = "b", marker = "o", s= 60)
plt.xlabel("Axb")
plt.ylabel("DWi")

#Durezas/potencia

plt.style.use('seaborn-whitegrid')
plt.scatter(x = DB1["BWi "], y =DB1["Potencia SAG01"], alpha = 0.7, c = "r", marker = "o", s=
60)
plt.xlabel("BWi")
plt.ylabel("Potencia SAG01 [kW]")

plt.style.use('seaborn-whitegrid')
plt.scatter(x = DB1["DWi"], y =DB1["Potencia SAG01"], alpha = 0.7, c = "r", marker = "o", s=
60)
plt.xlabel("DWi")
plt.ylabel("Potencia SAG01 [kW]")

plt.style.use('seaborn-whitegrid')
plt.scatter(x = DB1["Abx "], y =DB1["Potencia SAG01"], alpha = 0.7, c = "r", marker = "o", s=
60)
plt.xlabel("Axb")
plt.ylabel("Potencia SAG01 [kW]")

#Variables operacionales/Potencia

plt.style.use('seaborn-whitegrid')
fig, axs = plt.subplots(2, 3)

axs[0, 0].scatter(x = DB1["Tratamiento L1 (ton)", y =DB1["Potencia SAG01"], alpha = 0.7, c =
"g", marker = "o", s= 40)
axs[0, 0].set_title('Tratamiento / Potencia [kW]', fontsize = 10)

axs[0, 1].scatter(x = DB1["P80 L1"], y =DB1["Potencia SAG01"], alpha = 0.7, c = "g", marker =
"o", s= 40)
axs[0, 1].set_title('P80 [um] / Potencia [kW]', fontsize = 10)

axs[0, 2].scatter(x = DB1["% sólidos L1"], y =DB1["Potencia SAG01"], alpha = 0.7, c = "g",
marker = "o", s= 40)
axs[0, 2].set_title('% Sólidos / Potencia [kW]', fontsize = 10)

axs[1, 0].scatter(x = DB1["Veloc. Giro SAG01 (rpm)", y =DB1["Potencia SAG01"], alpha = 0.7, c
= "g", marker = "o", s= 40)
axs[1, 0].set_title('Velocidad de giro [rpm] / Potencia [kW]', fontsize =10)

axs[1, 1].scatter(x = DB1["Horas de operación L1"], y =DB1["Potencia SAG01"], alpha = 0.7, c =
"g", marker = "o", s= 40)
axs[1, 1].set_title('Horas de operación / Potencia [kW]', fontsize = 10)
```

```

# Modelo RLM
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import RepeatedKFold

X = np.array([DB1['BWl '], DB1['Tratamiento L1 (ton)'], DB1['P80 L1'], DB1['% sólidos L1'],
DB1['Veloc. Giro SAG01 (rpm)']])
X = X.transpose()
y = np.array([DB1['Potencia SAG01']])
y = y.transpose()

# Definir el número de divisiones (folds)
num_folds = 3
# Inicializar el objeto KFold
kf = RepeatedKFold(n_splits=num_folds, n_repeats=100)
# Lista para almacenar los errores cuadráticos medios de cada fold
mse_scores = []
r2_scores = []

# Iterar sobre los folds
for train_index, test_index in kf.split(X):
    # Dividir los datos en conjuntos de entrenamiento y prueba
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]
    # Crear y entrenar el modelo de regresión lineal
    modelo = LinearRegression()
    modelo.fit(X_train, y_train)
    # Realizar predicciones en el conjunto de prueba
    predicciones = modelo.predict(X_test)
    # Calcular el error cuadrático medio
    mse = mean_squared_error(y_test, predicciones)
    mse_scores.append(mse)
    r2 = r2_score(y_test, predicciones)
    r2_scores.append(r2)

# Calcular el promedio de los errores cuadráticos medios de todos los folds
mse_promedio = np.mean(mse_scores)
rmse_promedio = mse_promedio**0.5
r2_promedio = np.mean(r2_scores)
# Imprimir los errores cuadráticos medios de cada fold y el promedio
print("Error Cuadrático Medio Promedio:", mse_promedio)
print("Raíz del Error Cuadrático Medio Promedio:", rmse_promedio)
print("r2 Medio:", r2_promedio)
print("Coeficientes:", modelo.coef_)
print("Término independiente:", modelo.intercept_)

#Graficar y/ypred
plt.style.use('seaborn-whitegrid')
plt.scatter(x = y_test, y = predicciones, alpha = 0.6, c = "b", marker = "o", s= 60)
plt.plot([y_train.min(), y_train.max()], [y_train.min(), y_train.max()], 'k--', lw=2)
plt.xlabel("Valor real")
plt.ylabel("Valor predicho")

# Modelo KNN

# Normalizar datos
def mean_norm(df_input):
    return df_input.apply(lambda x: (x - x.mean()) / x.std(), axis=0)
DB1n = mean_norm(DB1)
def mean_denorm(df_input):
    return df_input.apply(lambda x: ((x * DB1["Potencia SAG01"].std())+ DB1["Potencia SAG01"].mean() ), axis=0)

from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsRegressor
from sklearn.metrics import mean_squared_error, r2_score

```

```

from sklearn.model_selection import RepeatedKFold

X = np.array([DBln['BWi '], DBln['Tratamiento L1 (ton)'], DBln['P80 L1'], DBln['% sólidos L1'], DBln['Veloc. Giro SAG01 (rpm)']])
X = X.transpose()
y = np.array([DBln['Potencia SAG01']])
y = y.transpose()

# Definir el número de divisiones (folds)
num_folds = 3
# Inicializar el objeto KFold
kf = RepeatedKFold(n_splits=num_folds, n_repeats=100)
# Lista para almacenar los errores cuadráticos medios de cada fold
mse_scores = []
r2_scores = []

# Iterar sobre los folds
for train_index, test_index in kf.split(X):
    # Dividir los datos en conjuntos de entrenamiento y prueba
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]
    # Crear y entrenar el modelo KNN
    modelo = KNeighborsRegressor(n_neighbors=8)
    modelo.fit(X_train, y_train)
    # Realizar predicciones en el conjunto de prueba
    predicciones = modelo.predict(X_test)
    # Calcular el error cuadrático medio
    y_test = pd.DataFrame(y_test)
    predicciones = pd.DataFrame(predicciones)
    y_test = mean_denorm(y_test)
    predicciones = mean_denorm(predicciones)
    mse = mean_squared_error(y_test, predicciones)
    mse_scores.append(mse)
    r2 = r2_score(y_test, predicciones)
    r2_scores.append(r2)

# Calcular el promedio de los errores cuadráticos medios de todos los folds
mse_promedio = np.mean(mse_scores)
rmse_promedio = mse_promedio**0.5
r2_promedio = np.mean(r2_scores)
# Imprimir los errores cuadráticos medios de cada fold y el promedio
print("Error Cuadrático Medio Promedio:", mse_promedio)
print("Raíz del Error Cuadrático Medio Promedio:", rmse_promedio)
print("r2 Medio:", r2_promedio)

#Graficar y/ypred
plt.style.use('seaborn-whitegrid')
plt.scatter(x = y_test, y = predicciones, alpha = 0.6, c = "b", marker = "o", s= 60)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--', lw=2)
plt.xlabel("Valor real")
plt.ylabel("Valor predicho")

# Modelo Random Forest
from sklearn.ensemble import RandomForestRegressor
from sklearn.datasets import make_regression
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import mean_squared_error

X = np.array([DBl['BWi '], DBl['Tratamiento L1 (ton)'], DBl['P80 L1'], DBl['% sólidos L1'], DBl['Veloc. Giro SAG01 (rpm)']])
X = X.transpose()
y = DBl['Potencia SAG01']

# Dividir los datos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)

```

```

# Definir los parámetros a buscar
parametros = {
    'n_estimators': [55],
    'max_depth': [14],
    'min_samples_split': [13],
    'min_samples_leaf': [1]
}

# Crear el modelo de Random Forest
modelo = RandomForestRegressor(random_state=1)

# Realizar la búsqueda de los mejores parámetros utilizando GridSearchCV
grid_search = GridSearchCV(modelo, parametros, cv=5, scoring='neg_mean_squared_error')
grid_search.fit(X_train, y_train)

# Obtener los mejores parámetros encontrados
mejores_parametros = grid_search.best_params_
print("Mejores Parámetros:", mejores_parametros)

# Crear el modelo con los mejores parámetros
modelo_final = RandomForestRegressor(**mejores_parametros, random_state=1)
# Entrenar el modelo con los datos de entrenamiento
modelo_final.fit(X_train, y_train)
# Realizar predicciones en el conjunto de prueba
predicciones = modelo_final.predict(X_test)

# Calcular el error cuadrático medio
mse = mean_squared_error(y_test, predicciones)
rmse = mse**0.5
r2 = r2_score(y_test, predicciones)
print("Error Cuadrático Medio:", mse)
print("Raíz del Error Cuadrático Medio Promedio:", rmse)
print("r2 Medio:", r2)

#Graficar y/ypred
plt.style.use('seaborn-whitegrid')
plt.scatter(x = y_test, y = predicciones, alpha = 0.6, c = "b", marker = "o", s= 60)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--', lw=2)
plt.xlabel("Valor real")
plt.ylabel("Valor predicho")

# Modelo XGBoost
import xgboost as xgb
from sklearn.datasets import make_regression
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import mean_squared_error

X = np.array([DB1['BWi '], DB1['Tratamiento L1 (ton)'], DB1['P80 L1'], DB1['% sólidos L1'],
DB1['Veloc. Giro SAG01 (rpm)']])
X = X.transpose()
y = DB1['Potencia SAG01']

# Dividir los datos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1)

# Definir los parámetros a buscar
parametros = {
    'learning_rate': [0.1],
    'max_depth': [2],
    'n_estimators': [120],
    'subsample': [0.5],
    'colsample_bytree': [0.8]
}

# Crear el modelo de XGBoost
modelo = xgb.XGBRegressor()

# Realizar la búsqueda de los mejores parámetros utilizando GridSearchCV

```

```
grid_search = GridSearchCV(modelo, parametros, cv=5, scoring='neg_mean_squared_error')
grid_search.fit(X_train, y_train)
```

```
# Obtener los mejores parámetros encontrados
mejores_parametros = grid_search.best_params_
print("Mejores Parámetros:", mejores_parametros)
```

```
# Crear el modelo con los mejores parámetros
modelo_final = xgb.XGBRegressor(**mejores_parametros)
# Entrenar el modelo con los datos de entrenamiento
modelo_final.fit(X_train, y_train)
# Realizar predicciones en el conjunto de prueba
predicciones = modelo_final.predict(X_test)

# Calcular el error cuadrático medio
mse = mean_squared_error(y_test, predicciones)
rmse = mse**0.5
r2 = r2_score(y_test, predicciones)
print("Error Cuadrático Medio:", mse)
print("Raíz del Error Cuadrático Medio Promedio:", rmse)
print("r2 Medio:", r2)
```

```
#Graficar y/ypred
plt.style.use('seaborn-whitegrid')
plt.scatter(x = y_test, y = predicciones, alpha = 0.6, c = "b", marker = "o", s= 60)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--', lw=2)
plt.xlabel("Valor real")
plt.ylabel("Valor predicho")
```

#GRAFICOS DE TIEMPO PRED/REAL

```
#Dividir conjunto de datos
filas = DB1.shape[0]
frac_filas = round(filas*7/10)
DB1_1 = DB1.iloc[:frac_filas]
DB1_2 = DB1.iloc[frac_filas:]
```

```
X = np.array([DB1_1['BWi '], DB1_1['Tratamiento L1 (ton)'], DB1_1['P80 L1'], DB1_1['% sólidos L1'], DB1_1['Veloc. Giro SAG01 (rpm)']])
X_train = X.transpose()
y_train = DB1_1['Potencia SAG01']
```

```
X = np.array([DB1_2['BWi '], DB1_2['Tratamiento L1 (ton)'], DB1_2['P80 L1'], DB1_2['% sólidos L1'], DB1_2['Veloc. Giro SAG01 (rpm)']])
X_test = X.transpose()
y_test = DB1_2['Potencia SAG01']
```

```
# Crear el modelo con los mejores parámetros
modelo_final = RandomForestRegressor(n_estimators = 55, max_depth = 14, min_samples_leaf = 1, min_samples_split = 13, random_state=1)
# Entrenar el modelo con los datos de entrenamiento
modelo_final.fit(X_train, y_train)
# Realizar predicciones en el conjunto de prueba
predicciones = modelo_final.predict(X_test)
```

```
# Crear el gráfico de líneas
x = DB1_2.index
y1 = y_test
y2 = predicciones

plt.plot(x, y1, color='tab:blue', label='Real')
plt.plot(x, y2, color='tab:orange', label='Estimada')
plt.xticks([0, 15, 30, 45, 60, 75, 90, 105, 120, 135, 146])
```

```
# Agregar etiquetas y título
plt.xlabel('Tiempo')
plt.ylabel('Potencia [kW]')
```

```
plt.title('Potencia estimada vs Potencia real')
plt.legend()
# Mostrar el gráfico
plt.show()
```