DESARROLLO DE UN MICROSERVICIO QUE PERMITA REALIZAR LA PREDICCIÓN DE ROP PARA UNA FORMACIÓN ROCOSA ESPECÍFICA

NICOLÁS CASTAÑO CARDONA

MARÍA FERNANDA SANTOS FRANCO

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE INGENIERÍAS FÍSICO-QUÍMICAS

ESCUELA DE INGENIERÍA DE PETRÓLEOS

BUCARAMANGA

2021

DESARROLLO DE UN MICROSERVICIO QUE PERMITA REALIZAR LA PREDICCIÓN DE ROP PARA UNA FORMACIÓN ROCOSA ESPECÍFICA

NICOLÁS CASTAÑO CARDONA

MARÍA FERNANDA SANTOS FRANCO

Trabajo de grado para optar al título de Ingeniero de Petróleos

Director:

Wilson Raul Carreño Velasco

M.Sc. en Diseño, Gestión y Dirección de Proyectos

Co-directores:

Diego Andrés Ojeda Vargas

M.Sc. en Analytics

Abraham Camilo Montes Humanez

M.Sc. en Ciencia Computacional: Inteligencia Artificial

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE INGENIERÍAS FÍSICO-QUÍMICAS

ESCUELA DE INGENIERÍA DE PETRÓLEOS

BUCARAMANGA

2021

DEDICATORIA

En primer lugar, agradezco a Dios, por sus infinitas bendiciones y por permitirme llegar a este momento, por la culminación de esta etapa y por las que vendrán.

A mis papás, Angel y Gloria y mi hermano Jesús, por ser el mayor pilar de mi vida, por ser mi fortaleza, mi paz y mis guías. Gracias a mi papá por su motivación, amor incondicional y alegría. Gracias a mi mamá por ser la luz de mi vida, mi refugio y ejemplo a seguir. Y a mi hermano por su paciencia, compañía y complicidad. Este logro es por ustedes, porque son mi principal motor y la razón de querer ser mejor y seguir adelante.

A mi amiga Olguita Martínez, que ya no está en este plano terrenal, también quiero dedicarle este logro y agradecerle por brindarme siempre su apoyo incondicional, por sus consejos y porque su paz y energía aportaron cosas maravillosas en mi vida y dejaron huella en mi corazón para siempre.

A todos mis amigos y compañeros, que hicieron de este proceso algo inolvidable y especial, gracias a su compañerismo, motivación, enseñanzas y apoyo.

A los ingenieros Wilson, Diego y Abraham por su disposición, por apoyarnos con su conocimiento y orientación, porque nos permitieron desarrollar una visión más amplia en la construcción del enfoque de nuestra carrera.

A Nico, mi gran amigo y compañero de tesis, por compartir esta experiencia conmigo, por su paciencia, motivación, por su apoyo tanto personal como académico y por cubrir de buena energía y de alegría todos los momentos.

Gracias a todos ustedes, porque hicieron de esta una excelente y enriquecedora experiencia.

María Fernanda Santos Franco

DEDICATORIA

Este libro es dedicado en primer lugar a mis padres Andrés Castaño y Diana Cardona quienes además de ser mi apoyo en lo mejores y peores momentos de mi vida, me han regalado siempre comprensión, amor, paciencia, la mejores palabras y consejos, pero lo más importante es que me apoyaron en todos mis sueños y metas como si fueran suyos sin importar que tan adversas fueran las circunstancias. Esto es por ustedes.

A mis abuelos por siempre preocuparse y estar conmigo siempre que los necesitaba, por acompañarme y estar presentes en cada paso.

A mi familia por siempre regalarme una voz de ánimo.

A mis compañeros, amigos y personas que conocí a lo largo de mi carrera que me permitieron aprender de ellos y que ahora son mi fuente de admiración, entusiasmo y gratitud.

A los ingenieros Wilson, Diego y Abraham no solo por darnos la oportunidad de aprender junto a ellos y estar siempre dispuestos a compartirnos sus conocimientos de la mejor manera, sino también por permitirnos enamorarnos cada día más de nuestra carrera.

A mi compañera de tesis Mafe, por ser más que mi compañera de tesis, una gran amiga que jamás dejó desvanecer su optimismo y siempre me contagió de la mejor energía y buena onda posible empujándome a ser mejor y siempre con una sonrisa.

A todos ellos gracias totales.

Nicolás Castaño Cardona

AGRADECIMIENTOS

Los autores expresan sus agradecimientos:

A nuestras familias, por acompañarnos y apoyarnos en cada momento de nuestras vidas y muy especialmente en esta etapa en particular.

A los ingenieros Wilson Carreño, Diego Ojeda y Abraham Montes por la confianza, disposición, acompañamiento, consejos y ayuda profesional brindada desde el principio hasta el final de este proceso.

A profesores, compañeros, directivos y a todas las personas que aportaron a nuestra formación personal y profesional durante todo nuestro ciclo universitario.

A la Universidad Industrial de Santander y a la Escuela de Ingeniería de Petróleos por permitirnos enriquecer cada día nuestro conocimiento en su claustro universitario y poder ser parte de tan prestigiosa institución.

CONTENIDO

pág.

LISTA DE FIGURAS

pág.

LISTA DE TABLAS

pág.

LISTA DE ANEXOS

pág.

LISTA DE ABREVIATURAS

Bbl: Barriles

degF: Grados Fahrenheit

ET: Extra Trees Regressor

Ft: (Feet) Pies

GPM: Galones por minuto

HKLD: (Hook load) carga en el gancho.

Hr: Horas

Klbf: Kilo libra fuerza

KNN: K Nearest Neighbor

LightGBM: Light Gradient Boosting Machine

MW: (Mud Weight) densidad del lodo

Ppg: (Pounds per galon) Libras por galón

Pump: Presión de la bomba de perforación

RF: Random Forest

ROP: (Rate of penetration) Tasa de penetración

RPM: Revoluciones por minuto

WOB: (Weight on bit) Peso sobre la broca

XGBoost: Extreme Gradient Boosting

RESUMEN

TITULO: DESARROLLO DE UN MICROSERVICIO QUE PERMITA REALIZAR LA PREDICCIÓN DE ROP PARA UNA FORMACIÓN ROCOSA ESPECÍFICA[*]

AUTOR: NICOLÁS CASTAÑO CARDONA, MARÍA FERNANDA SANTOS FRANCO[**]

PALABRAS CLAVE: ROP, PERFORACIÓN, ANALÍTICA PREDICTIVA, INTELIGENCIA ARTIFICIAL, MACHINE LEARNING, MICROSERVICIO.

DESCRIPCIÓN: Debido a la necesidad de optimizar el desempeño de las operaciones de perforación con la estimación precisa de parámetros como la ROP y gracias a la disponibilidad de nuevas tecnologías basadas en Analytics, se desarrolla un microservicio que permite realizar la predicción de ROP para una formación rocosa específica caracterizada por ser dura y abrasiva.

Inicialmente se presentó una revisión de antecedentes para comprender y valorar los precedentes en la predicción de la ROP. Después, se realizó el análisis exploratorio, procesando los datos en estado dinámico de distintas corridas en la formación objeto de estudio, que permitió explorar la distribución identificando valores atípicos y concentraciones en los datos que contribuyen con la limpieza y preparación de los datos. Luego, se procedió a realizar el análisis diagnóstico de las variables encontrando patrones y correlaciones. Con estos resultados se analiza el comportamiento de las variables de entrada y su incidencia en la ROP, además de formar una base de datos compacta y lista para el modelamiento.

Esta base de datos alimenta los modelos de machine learning, utilizando las librerías de Python para conocer los modelos que mejor se ajustan a los datos, procediendo a elegir tres de ellos para su implementación (KNN, LightGBM y XGBoost). Se crean los modelos seleccionados ajustando sus hiperparámetros para luego evaluar la eficiencia de la métrica del coeficiente de determinación de cada modelo, encontrando así que el modelo con mayor desempeño es el XGBoost con un 74% de R2. Finalmente, se realiza el deployment eligiendo el modelo con el mejor desempeño en la predicción y que cumpla con la capacidad de la plataforma como servicio, Heroku, para crear el aplicativo web que permite realizar la predicción de la ROP en la formación de estudio, convirtiendo la información de los datos históricos en ideas simples y procesables.

[*] Trabajo de grado

[**] Facultad de Ingenierías Físico-Químicas, Escuela de Ingeniería de Petróleos. Director: Wilson Raul Carreño Velasco, M.Sc. en Diseño, Gestión y Dirección de Proyectos. Codirectores: Diego Andrés Ojeda Vargas, M.Sc. en Analytics, Abraham Camilo Montes Humanez, M.Sc. en Ciencia Computacional: Inteligencia Artificial.

ABSTRACT

TITLE: DEVELOPMENT OF A MICROSERVICE THAT ALLOWS TO PERFORM ROP PREDICTION FOR A SPECIFIC ROCK FORMATION [*]

AUTHOR: NICOLÁS CASTAÑO CARDONA, MARÍA FERNANDA SANTOS FRANCO[**]

KEY WORDS: ROP, DRILLING, PREDICTIVE ANALYTICS, ARTIFICIAL INTELLIGENCE, MACHINE LEARNING, MICROSERVICE.

DESCRIPTION: Due to the need to optimize the performance of drilling operations with the precise estimation of parameters such as ROP and thanks to the availability of new technologies based on Analytics, a microservice is developed that allows the prediction of ROP for a specific rock formation characterized by being hard and abrasive.

Initially, a background check was presented to understand and assess the precedents in the prediction of ROP. Afterwards, the exploratory analysis was carried out, processing the data in the dynamic state of different runs in the formation under study, which allowed exploring the distribution by identifying atypical values and concentrations in the data that contribute to the cleaning and preparation of the data. Then, the diagnostic analysis of the variables was carried out, finding patterns and correlations. With these results, the behavior of the input variables and their impact on ROP are analyzed, in addition to forming a compact database ready for modeling.

This database feeds the machine learning models, using Python libraries to find out the models that best fit the data, proceeding to choose three of them for their implementation (KNN, LightGBM and XGBoost). The selected models are created by adjusting their hyperparameters to then evaluate the efficiency of the metric of the determination coefficient of each model, thus finding that the model with the highest performance is the XGBoost with 74% of R2. Finally, the deployment is carried out choosing the model with the best performance in the prediction and that complies with the capacity of the platform as a service, Heroku, to create the web application that allows the prediction of the ROP in the formation of study, converting information from historical data into simple, actionable ideas.

[*] Degree Work

[**] Faculty of Physicochemical Engineering, School of Petroleum Engineering. Director: Wilson Raul Carreño Velasco, M.Sc. Design, Management and Project Management. Codirectors: Diego Andrés Ojeda Vargas, M.Sc. Analytics, Abraham Camilo Montes Humanez, M.Sc. Computer Science: Artificial Intelligence.

INTRODUCCIÓN

En gran medida, el principal objetivo dentro de cualquier industria es poder proporcionar a sus clientes servicios de alta calidad con un menor costo. En los últimos años, este objetivo ha venido siendo cada vez más alcanzable debido a las nuevas tecnologías emergentes, entre ellas particularmente el aprendizaje de máquinas o Machine Learning.[1] La industria petrolera a pesar de estar ubicada entre uno de los sectores que más volúmenes de datos genera, estos datos no están siendo particularmente útiles. Según Mark Spelman et al.[2] aproximadamente el 36% de las empresas de Oil & gas están invirtiendo en Big Data y Analítica, sin embargo, solo el 13% utiliza los conocimientos adquiridos en la aplicación de esta tecnología, mostrando así que la mayoría de las empresas no han incorporado esta información en sus sistemas con el fin de mejorar su productividad en las operaciones.

Si bien el sector petrolero no es el más avanzado en inteligencia artificial, su implementación es inminente y las aplicaciones predictivas del machine learning en esta son muchas, desde la identificación de fracturas y/o regiones a perforar, de manera que se produzca la mayor cantidad de hidrocarburos al menor costo, mantenimiento preventivo de bombas electro sumergibles o mecánicas, hasta la predicción de la tasa de penetración ROP en pozos y detección de procesos de parafinación que reduzcan los volúmenes de petróleo producido.[3]

Cuando se realiza la perforación de un pozo para la extracción de hidrocarburos, es fundamental conocer el comportamiento de diferentes parámetros que son medidos durante el proceso, entre ellos se establece la ROP como uno de los más importantes para poder llegar a un óptimo rendimiento, reducción de costos y tiempo

[1] MAISUECHE CUADRADO, Alberto. Utilización del Machine Learning en la industria 4.0. Universidad de Valladolid. 2019.
[2] SPEALMAN, Mark, et al. Digital Transformation Initiative Oil and Gas Industry. World Economic Forum. 2017.
[3] ESTANISLAO, Irigoyen. Industria 4.0 y transformación digital en la industria del petróleo y gas: Situación actual. Petrotecnica. 2019.

de perforación. Teniendo en cuenta lo anterior, durante el proceso de perforación se atraviesan distintas formaciones, algunas de ellas son duras y abrasivas, y en algunos casos no son de interés e incluso pueden llegar a 800 ft de espesor por la trayectoria de los pozos y la incertidumbre en cuanto a los buzamientos, por lo tanto, el reto es conseguir que las corridas sean largas y con óptima ROP lo cual reduce la cantidad de viajes de tubería.[4]

En la actualidad, gran parte de los modelos utilizados para la predicción de la ROP son considerados tradicionales, en donde solo se tienen en cuenta un número limitado de parámetros, razón por la cual, pueden disminuir su efectividad en entornos complicados como los mencionados anteriormente. Por estas razones, esta investigación tiene el propósito de estudiar, analizar y predecir el comportamiento de la tasa de penetración en una formación dura y abrasiva, mediante el uso de algoritmos de machine learning supervisados como K Nearest Neighbors, Light Gradient Boosting Machine y Extreme Gradient Boosting y, posteriormente, realizar el despliegue de un microservicio web, donde se aglutinarán todos los resultados obtenidos con el fin de realizar un predictor eficaz y de fácil acceso para cualquier usuario.

[4] MONTES, Abraham, CARREÑO, Wilson y GUÍO, Miguel. Aspectos de la perforación de pozos complejos en piedemonte en tiempos de crisis. El reventón energético. 2018.

1. OBJETIVOS

1.1. OBJETIVO GENERAL

Desarrollar un microservicio que permita realizar la predicción de la ROP para una formación rocosa específica.

1.2. OBJETIVOS ESPECÍFICOS

- Realizar una revisión de literatura sobre el uso de analítica predictiva con ROP.
- Efectuar el análisis exploratorio de los datos objeto de estudio.
- Realizar un análisis diagnóstico de las variables y el impacto de éstas en la predicción de la ROP.
- Realizar un análisis predictivo de la ROP de una formación a partir de la selección y prueba de algoritmos de inteligencia artificial.

2. MARCO TEÓRICO

Para lograr una asimilación de los conceptos que sirva de guía en el entendimiento del presente proyecto, se inicia con una presentación de antecedentes, investigaciones anteriores relacionadas con el objeto de investigación actual, realizadas por diferentes autores, pasando enseguida a una revisión de conceptos de análisis de datos y, finalmente enfocar una ambientación técnica que servirá para dar una mejor comprensión del problema y de los diferentes parámetros que se van a emplear en este trabajo.

2.1. ANTECEDENTES

Actualmente, los conceptos de Big Data, la ciencia de datos, inteligencia artificial, hacen parte de los más utilizados cuando se refiere a la industria 4.0, una cuarta revolución enfocada en la automatización y al intercambio de datos. La información se genera y almacena en cantidades que habrían sido inimaginables en décadas anteriores. Hasta el año 2003, la humanidad había almacenado una cantidad de 5 Exabytes de datos digitales, entre 2003 y 2012, una cantidad de 13.7 Exabytes de datos nuevos[5], sin embargo, solo en 2018 se generaron alrededor de 33 Zettabits de información, que en unidades de datos, equivale aproximadamente a 1.000 millones de Terabytes, es decir, alrededor de cinco mil veces más que todos los datos recogidos por la humanidad hasta el año 2012.[6]

Específicamente en la industria petrolera, a pesar de ser uno de los sectores que más tiempo ha llevado la vanguardia tecnológica en los últimos años, y estar dentro

[5] CHAPARRO, Mauricio. Factores críticos para la implementación de proyectos que utilizan datos masivos Big Data en organizaciones operadoras de la industria del petróleo y gas en Colombia industria del petróleo y gas en Colombia. 2021.
[6] MORENO, A. A la espera de un Big Bang de datos. [Online]. 2019.

de las industrias que más volúmenes de información generan, su adaptabilidad respecto a estas tecnologías es lenta, ya que sigue siendo una industria intuitiva y experimental cuando de tomar decisiones se trata.[7] Sin embargo, hacia el año 2019, la inteligencia artificial en la industria Oil & Gas tenía un valor de dos mil millones de dólares, un valor ya considerable que espera un crecimiento del 10.96% hacia el año 2025, un aumento que aspira llegar a los 3.81 mil millones de dólares.[8]

Teniendo en cuenta lo anterior, las empresas de Oil & Gas están haciendo uso de machine learning para probar qué impacto puede tener un proyecto concreto o tener conocimiento de los riesgos ambientales que pueden derivarse del mismo, así como monitoreo de procesos de perforación y responder a un problema de una manera más rápida de lo que la capacidad humana lo permite.[9] Este proyecto de grado tiene como objeto de estudio la predicción de la tasa de perforación que a partir de este punto se seguirá llamando ROP por sus siglas en inglés (Rate of penetration), esta predicción ha sido objeto de estudio por autores anteriores ya que es un factor determinante en la etapa de perforación de un pozo petrolífero.

Barbosa et. al.[10] en el año 2019, realizó un compilado de 53 investigaciones basadas en la predicción de la ROP usando diferentes modelos de machine learning. En su investigación se denota una preferencia absoluta por modelos neuronales, más conocidos como ANN (artificial neural network), en 18 de los 53 artículos se realiza una comparación de eficiencia de los modelos computacionales con modelos tradicionales como el modelo de Bourgoyne y Young. En 17 de estos artículos, los modelos de machine learning consiguieron resultados más eficientes debido a su capacidad de capturar modelos no lineales, solo en uno de ellos

[7] CHAPARRO, Op. Cit.
[8] RAVIOSA, Alexandro. Inteligencia Artificial, apuesta obligada para la industria del petróleo. Global Energy [Online], 2020.
[9] Ibid.
[10] BARBOSA, Luís Felipe, et al. Machine learning methods applied to drilling rate of penetration prediction and optimization - A review. 2019.

concluyó que no había una diferencia notable, sin embargo, una abrumadora superioridad por parte de los modelos de aprendizaje automático.

Con respecto a los modelos usados, 39 de ellos se decidieron por realizar una predicción con modelos neuronales, a pesar de ser modelos con una buena capacidad de predicción, son usualmente más complejos, a estos modelos se le suele denominar "black-boxes" o caja negra, ya que son particularmente difícil de comprender y explicar[11]. Debido a ello, en este proyecto se decidió atender a diferentes alternativas para la realización del mismo, particularmente modelos de ensemble como lo son los modelos de boosting.

Por lo que se refiere a la actualidad colombiana, la cuarta revolución industrial también se hace presente, según Ernesto Gutiérrez[12], vicepresidente digital de la empresa de petróleos estatal colombiana Ecopetrol, se está usando machine learning y Deep learning para predecir y saber más de aspectos que no se conocen. Actualmente Ecopetrol se articula con la inteligencia artificial para ampliar sus conocimientos y tomar decisiones más convenientes. Con base en esto, diversidad de investigaciones se han hecho en el país usando inteligencia artificial. En 2021, Nathalia Tabares y Daniel Tovar[13] realizaron un proyecto en el que utilizando machine learning desarrollaron un modelo predictivo para la estimación de ROP y MSE (energía mecánica especifica) en el campo Yarigui - Cantagallo de la empresa Ecopetrol.

En relación con los modelos usados en su investigación, los autores decidieron en primera instancia recurrir directamente al uso de un modelo de machine learning de aprendizaje supervisado Random forest haciendo uso del lenguaje de programación Python[14], mismo que será utilizado en el presente proyecto de grado debido a su

[11] BARBOSA. Op. Cit.
[12] SYRUS. Usos de la inteligencia artificial en la industria petrolera. Syrus España [Online]. 2020.
[13] RODRIGUEZ, Nathalia y TOBAR, Daniel. Implementación De Un Modelo Predictivo De Machine Learning Para La Estimación De Los Parámetros Óptimos De La Rop Y La Mse En La Sección 8½'' Y 12 ¼'' Para Los Pozos Perforados Con Motor De Fondo En El Campo Yarigui – Cantagallo Durante El 2019. 2021.
[14] Ibid.

versatilidad al ser un lenguaje multiplataforma, además de ser un software gratuito y de uso libre especializado en modelos de aprendizaje automático y en procesamiento de grandes cantidades de datos.[15]

De acuerdo con lo anterior, la inteligencia artificial en Colombia es un sector emergente con grandes proyecciones especialmente en la industria petrolera. En la presente investigación se busca aportar al conocimiento de estas técnicas de predicción contemporáneas realizando un proyecto end–to–end de machine learning, haciendo una comparativa entre diferentes modelos, eligiendo el más conveniente según el presupuesto, cantidad de datos, software disponible, nivel de complejidad y diferentes parámetros que serán vistos a través de este documento, además de finalmente realizar una puesta en producción a través de un microservicio con los resultados obtenidos, usando librerías como streamlit de Python y haciendo uso de servicios de plataforma administrada gratuita como Heroku que permitirán que cualquier persona pueda acceder y realizar predicciones con datos personalizados, para poder sacar el máximo provecho a los resultados obtenidos en este proyecto.

2.2. MACHINE LEARNING

El aprendizaje de máquinas o machine learning ha venido siendo durante los últimos años uno de los principales temas acerca de revolución tecnológica. Desde 1943, cuando el matemático Walter Pitts[16] acuñó por primera vez el término “Inteligencia artificial” proponiendo que el cerebro funciona como una computadora que utiliza las neuronas para procesar información, hasta la actualidad diversos personajes de la humanidad han contribuido de manera acertada a la comprensión de los

[15] GAGO, Ultrario. Uso de algoritmos de aprendizaje automático aplicados a bases de datos genéticos. Universidad Oberta de Catalunya. Cataluña. 2017.
[16] HINESTROZA, Denniye. El Machine Learning A Través De Los Tiempos, Y Los Aportes A La Humanidad. Universidad Libre Seccional Pereira. Pereira. 2018.

conceptos que competen al aprendizaje automático. Machine learning es un subcampo de la computación cuyo objetivo es desarrollar técnicas que permitan a los computadores aprender, es decir, generalizar comportamientos a partir de información suministrada en forma de ejemplos.[17]

Consecuente con lo anterior, el machine learning se puede clasificar en diferentes categorías, entre ellas aprendizaje supervisado, no supervisado y reforzado.

En general, el aprendizaje supervisado o aprendizaje con clase, consiste en aprender un patrón a partir de un conjunto de datos que se utilizan para entrenar un modelo que permitirá predecir un conjunto de datos no observados anteriormente.[18] Es decir, en este tipo de aprendizaje se realiza una predicción usando datos conocidos para entrenar el modelo conociendo cual es la variable de salida deseada.

Otro concepto dentro de la clasificación del machine learning es el aprendizaje no supervisado o aprendizaje sin clase, en este se hace uso de datos que no tienen etiquetas ni estructura que permitan una fácil identificación de los patrones o clases a los que hace parte cada conjunto de datos, por lo tanto, el objetivo del aprendizaje sin clase es la creación de distintos subconjuntos a partir de los datos entregados inicialmente, y estos subgrupos se realizan teniendo en cuenta las similitudes de las características de los datos generando patrones determinados.[19]

Por último y no menos importante, se encuentra el aprendizaje reforzado, en este caso a diferencia de los demás, su objetivo es aprender a decidir mediante su propia experiencia, es decir, que cuando se presenta una situación específica pueda decidir cuál es la mejor elección respecto a ese caso a partir de un proceso de prueba y error basado en el refuerzo positivo cuando se logra un objetivo o está cerca de lograrlo.

---

[17] GAGO, Op. Cit. .
[18] ZAMORANO, Juan. Comparativa Y Análisis De Algoritmos De Aprendizaje Automático Para La Predicción Del Tipo Predominante De Cubierta Arbórea. Universidad Complutense de Madrid. 2018.
[19] Ibid.

En este proyecto de grado, particularmente se tienen datos etiquetados y estructurados cuyo objetivo es predecir una variable de salida que se encuentra bien definida (ROP), teniendo en cuenta lo anterior, en este trabajo se procederá con algoritmos y modelos enfocados hacia el aprendizaje supervisado o con clase.

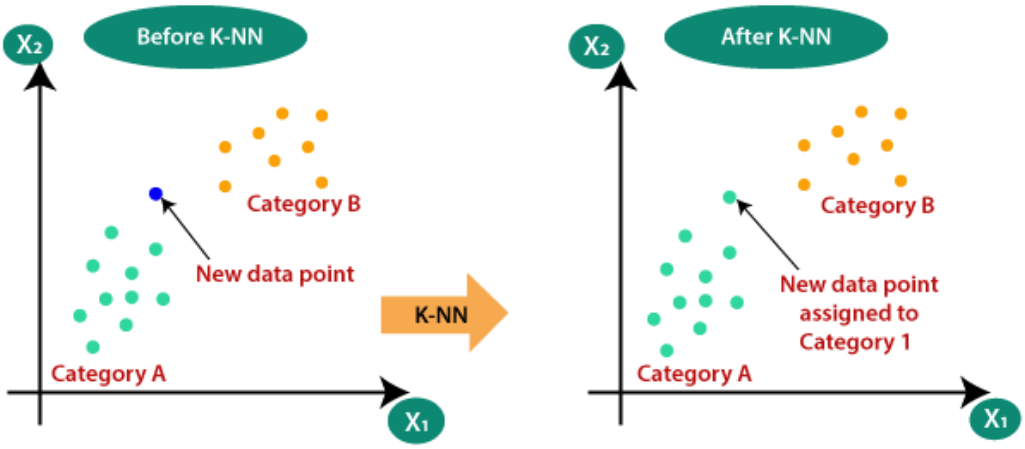
2.2.1. Algoritmos de clasificación supervisados. Luego de establecer el modelo de aprendizaje a usar, se procede a comprender algunos de los algoritmos más usados dentro del mundo del aprendizaje automático. Entre ellos el KNN (K Nearest Neighbors), árboles de decisión y métodos de ensemble como boosting.

2.2.1.1. KNN. El modelo K-Nearest Neighbors o K-Vecinos más cercanos se basa en la búsqueda de distancias entre una consulta y todos los valores en los datos, seleccionando el número especificado de ejemplos (K) más cercanos a la consulta, luego elige la etiqueta más frecuente (en el caso de la clasificación) o promedia las etiquetas (en el caso de la regresión). KNN aprende través de cada uno de los registros que se tengan en la base de datos y opera a través de la distancia euclidiana que para casos bidimensionales es el teorema de Pitágoras.[20]

La figura 1 representa el comportamiento del algoritmo de k vecinos más cercanos, teniendo en cuenta la distancia del punto azul a sus vecinos más próximos, se define que serían los datos de la categoría A, es decir, los puntos verdes.

[20] PETERSON, Leif. K-nearest neighbor. Scholarpedia. Scholarpedia [Online]. 2019.

Figura 1. Ejemplo de diagrama de K vecinos más próximos



Fuente*:* PETERSON, Leif. K-nearest neighbor. Scholarpedia. [Online]. 2019. http://scholarpedia.org/article/K-nearest_neighbor

Tabla 1. Fortalezas y debilidades del modelo Knn

Fortalezas	**Debilidades**
Simple y efectivo.	No produce un modelo, limitando la habilidad para entender cómo se relacionan las características con las clases.
No hace suposiciones sobre la distribución de los datos.	Requiere relacionar un valor k apropiado.
Fase de entrenamiento rápida.	Fase de clasificación lenta.
	Características nominales y datos perdidos requieren procesamiento adicional.

Fuente: ZAMORANO, Juan. Comparativa Y Análisis De Algoritmos De Aprendizaje Automático Para La Predicción Del Tipo Predominante De Cubierta Arbórea. Universidad Complutense de Madrid. 2018.

Zamorano[21] presenta en la tabla 1 con las fortalezas y debilidades de este modelo, mostrando así que a pesar de su sencillez en su modo de ejecución es uno de los más usados, sin embargo, muestra ciertas deficiencias y limitaciones.

2.2.1.2. Árboles de decisión. Los árboles de decisión hacen parte de los modelos predictivos supervisados, formado por una serie de reglas binarias, con las que se consigue repartir diferentes observaciones en función de sus atributos, cada input nuevo pasará por este proceso de clasificación para finalmente obtener un valor de la variable respuesta al final de las ramas del árbol.[22]

Al igual que en knn, Zamorano[23] plantea en la tabla 2 bajo los mismos criterios las ventajas y desventajas de este método, donde muestra que a pesar de ser un método fácil y simple de entender, en ocasiones puede generar árboles muy complejos que disminuyen calidad de predicción al modelo planteado.

Tabla 2. Fortalezas y debilidades del modelo árboles de decisión

Fortalezas	Debilidades
Simple de entender e interpretar. Se pueden visualizar los árboles construidos.	Se pueden generar árboles muy complejos, que no generalizan bien los datos.
No es necesario preparar mucho los datos.	Pequeñas variaciones en los datos producen grandes cambios en la estructura generada.

Fuente: ZAMORANO, Juan. Comparativa Y Análisis De Algoritmos De Aprendizaje Automático Para La Predicción Del Tipo Predominante De Cubierta Arbórea. Universidad Complutense de Madrid. 2018.

[21] ZAMORANO. Op. Cit.
[22] AMAT, Joaquín. Árboles de decisión, random forest , gradient boosting y C5.0. 2017.
[23] ZAMORANO. Op. Cit.

2.2.1.3. Métodos de ensemble. Estos métodos se basan en combinaciones de diferentes modelos en los cuales es necesario definir cómo se van a crear y de qué manera se van a combinar estos para obtener una mejor producción de cada uno de los modelos individualmente. Dentro de este método, uno de los más usados y caracterizado como uno de los mejores es el Boosting, en este cada modelo trata de solucionar errores del modelo anterior, es decir, en un primer modelo donde se realizan determinadas predicciones y se cometerán algunos errores, un segundo modelo entrará para corregir estos errores dándole un mayor peso a las muestras mal clasificadas y enfocándose en los resultados acertados.[24]

Dentro de la clasificación de boosting, existe un método llamado Gradient Boosting Decision Tree (GBDT), formado por un conjunto de árboles de decisión individuales que serán entrenados iterativamente de manera secuencial donde cada árbol corregirá los errores del árbol anterior, generando así una predicción final.[25] Entre estos métodos se encuentran los modelos de boosting más usados actualmente como el Extreme Gradient Boosting y Light Gradient Boosting Machine.

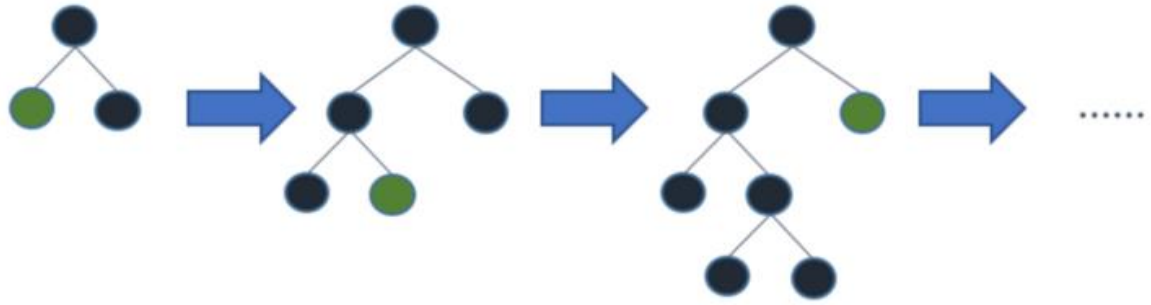
Light Gradient Boosting Machine se refiere al proyecto de código abierto, la biblioteca de software y el algoritmo de aprendizaje de la máquina. Este modelo amplía el algoritmo de aumento de gradientes añadiendo un tipo de selección automática de características, así como centrándose en aumentar los ejemplos con gradientes más grandes. Esto puede resultar en una dramática aceleración del entrenamiento y en una mejora del rendimiento predictivo. Esta clase de algoritmos de aprendizaje de máquinas en conjunto pueden utilizarse para la clasificación o la regresión de problemas de modelado predictivo.

La diferencia del modelo con otros algoritmos basados en árboles es que LightGBM hace crecer el árbol verticalmente, como se observa en la figura 2, mientras que

[24]IARTIFICIAL. Ensembles: voting, bagging, boosting, stacking.[Online] 2019.
[25]AMAT. Op. Cit.

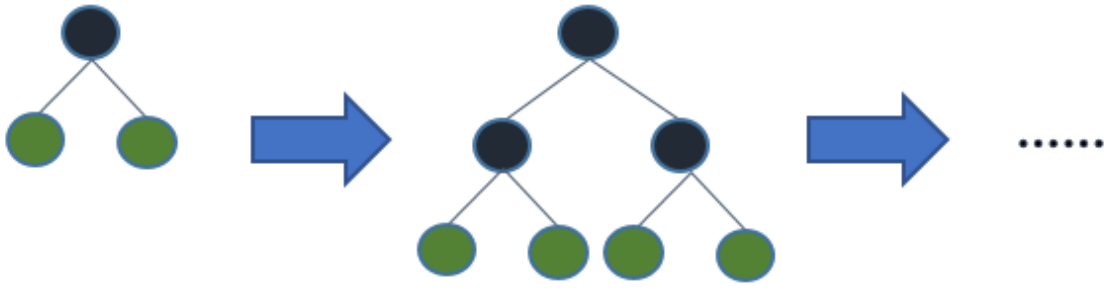
otro algoritmo hace crecer los árboles horizontalmente (figura 3), lo que significa que crece en forma de hojas de árbol mientras que otro algoritmo crece a nivel.[26]

Figura 2. Funcionamiento del modelo LightGBM



Fuente: KHANDELWAL, Pranjal. Which algorithm takes the crown. Analytics Vidhya [Online]. 2017. https://www.analyticsvidhya.com/blog/2017/06/which-algorithm-takes-the-crown-light-gbm-vs-xgboost/

Figura 3. Proceso del algoritmo del modelo XGBoost



Fuente*:* KHANDELWAL, Pranjal. Which algorithm takes the crown. Analytics Vidhya [Online]. 2017. https://www.analyticsvidhya.com/blog/2017/06/which-algorithm-takes-the-crown-light-gbm-vs-xgboost/

[26] FAN, Junliang, et. al. Light Gradient Boosting Machine: An efficient soft computing model for estimating daily reference evapotranspiration with local and external meteorological data. ScienceDirect. 2019

Extreme Gradient Boosting es una implementación de árboles de decisión con Gradient boosting, diseñada para minimizar la velocidad de ejecución y maximizar el rendimiento. Este modelo consiste en un ensamblado secuencial de árboles de decisión (este ensamblado se conoce como CART, acrónimo de “Classification and Regression Trees”). XGBoost utiliza procesamiento en paralelo, poda de árboles, manejo de valores perdidos y regularización (optimización que penaliza la complejidad de los modelos) para evitar en lo posible sobreajuste o sesgo del modelo.[27]

2.3. ESTADÍSTICA

Dentro de esta sección se expondrán diferentes conceptos estadísticos importantes a la hora de entender secciones posteriores del proyecto, conceptos como medidas de tendencia central y dispersión, histogramas, diagramas de caja y bigotes, correlaciones entre variables y finalmente métricas de evaluación para proyectos de machine learning.

2.3.1. Medidas de tendencia central y dispersión. Las medidas de tendencia central son medidas estadísticas que se enfocan en resumir un conjunto de valores en un solo número, estas presentan un centro frente al cual orbitan la mayoría de los datos. Por otra parte, las medidas de dispersión, como su nombre lo indica, miden qué tan dispersos o qué tanto difieren los valores entre sí, o respecto a un valor en particular. De esta manera, ambas medidas juntas permiten analizar un

[27] SHERIDAN, Robert. Extreme Gradient Boosting as a Method for Quantitative Structure–Activity Relationships.ASCpublication [Onlines]. 2016.

conjunto de datos de tal forma que se obtenga información útil acerca de su posición y dispersión.[28]

A continuación, se presentará en la tabla 3 la definición de cada una de las principales medidas de tendencia central y dispersión con su respectivo símbolo.

Tabla 3. Medidas de tendencia central y dispersión

Medida	Símbolo	Definición
Media Aritmética	μ	La media representa el valor central de los datos constituyendo ser la medida de ubicación que más se utiliza.
Mediana	Me	La mediana es el valor de la variable que ocupa la posición central, cuando los datos se disponen en orden de magnitud.
Moda	Mo	La moda de una distribución se define como el valor de la variable que más se repite.
Varianza	S^2	Es la medida que cuantifica la variabilidad de los datos respecto al valor de la media.
Desviación estándar	S	Es la raíz cuadrada positiva de la varianza. Mide la variabilidad de los datos en las unidades en que se midieron originalmente.

Fuente*:* Autores.

2.3.2. Distribución normal**.** La distribución normal es considerada como la más importante de todas las distribuciones de probabilidad, su importancia radica en que gran número de fenómenos naturales se pueden modelar con esta distribución,

[28] QUEVEDO, Fernando. Medidas de tendencia central y dispersión. Universidad de Chile. Santiago de Chile. 2011.

además, por el teorema del límite central, todas aquellas variables que puedan considerarse causadas por un gran número de pequeños efectos tienden a distribuirse con una distribución normal.[29]

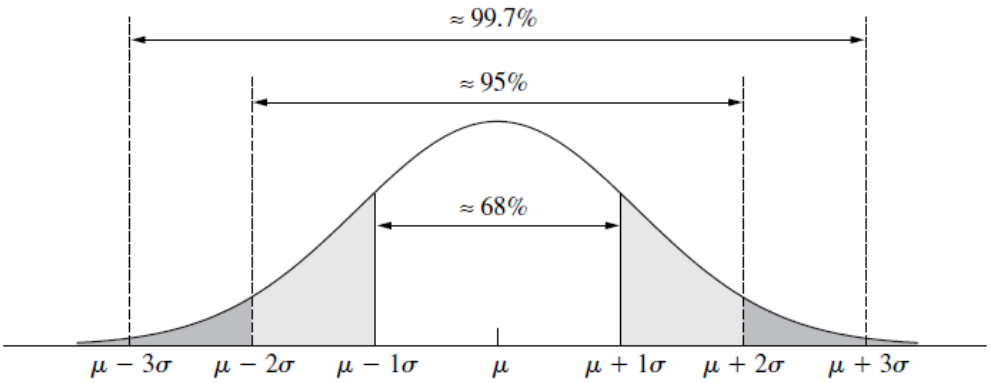
Este modelo teórico es capaz de aproximar satisfactoriamente el valor de una variable aleatoria a una situación ideal. La función densidad de probabilidad en una variable aleatoria normal con media μ y varianza S^2 que está dada por la siguiente fórmula[30]:

$$f(x) = \frac{1}{s\sqrt[2]{2 * \pi}} e^{-\frac{(x-\mu)^2}{2*S^2}}$$

La proporción de la población normal que está a cierto número de desviaciones estándar de la media suele ser la misma en cualquier población normal. De esta manera, se puede concluir que la mayoría de las poblaciones que cumplen con los parámetros normales suelen tener un porcentaje de 68% de la población aproximadamente en el intervalo entre μ ± s, 95% entre μ ± 2s y finalmente 99.7% de la población se encuentra en el intervalo entre μ ± 3s[31], como se muestra en la figura 4.

[29] LEJARZA, J. y LEGARZA, I. Distribución normal. 2018.
[30] NAVIDI, William. Estadística Para Ingenieros y Científicos. Mc Graw Hill. Colorado School of Mines. 2006.
[31] Ibid.

Figura 4. Gráfica de distribución normal

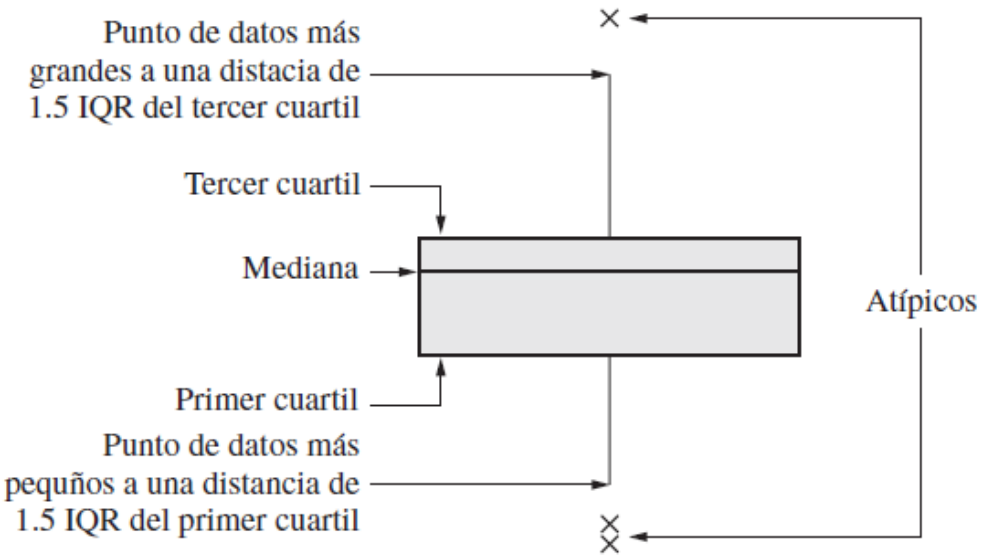


Fuente: NAVIDI, William. Estadística Para Ingenieros y Científicos. Mc Graw Hill. Colorado School of Mines. 2006.

2.3.3. Diagrama de caja y bigotes. Los diagramas de caja o también conocidos como diagramas de caja y bigotes son diagramas que incluyen la mediana, el primer y tercer cuartil y además los valores atípicos dentro de un conjunto de datos. Para poder entender este diagrama es necesario conocer el concepto de rango Inter cuartil (IQR) que representa la diferencia entre el primer y tercer cuartil, como se muestra en la figura 5, esto indica que, si el 25% de los datos se encuentra por debajo del primer cuartil y el 75% por debajo del tercer cuartil, entonces la diferencia entre ambos es donde se encuentra la mitad de los datos, es decir, el IQR. Los valores atípicos son aquellos valores que son inusualmente grandes o pequeños, dependiendo del conjunto de datos que se evalúa, en un diagrama de caja se muestran como aquellos valores que se encuentran a más de 1.5 IQR o menos de 1.5 IQR respecto al tercer y primer cuartil respectivamente.[32]

[32] NAVIDI. Op. Cit.

Figura 5. Distribución de un diagrama de caja



Fuente*:* NAVIDI, William. Estadística Para Ingenieros y Científicos. Mc Graw Hill. Colorado School of Mines. 2006.

2.3.4. Métricas de medición de modelos. Los modelos de Inteligencia artificial suelen enfocarse con el fin de resolver un problema o de predecir una posible variable en este caso, pero a su vez, estos deben ser comparados de cierta manera para determinar si el modelo evaluado está cumpliendo su función de manera efectiva, de allí deriva la importancia de conocer las métricas de medición y conocer que tan factible es que la variable evaluada arroje resultados satisfactorios. A continuación, se presentarán las métricas más usadas en modelos de machine learning las cuales servirán para evaluar el desempeño de los modelos de este proyecto.

Error absoluto medio (MAE). El MAE es una medida que entrega como resultado la cercanía entre la predicción hecha con el valor real, esto partiendo de una

diferencia entre el valor obtenido y el valor absoluto de la medición real, para luego calcular el promedio.[33] Se describe mediante la siguiente ecuación:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|yi - \hat{y}i|$$

Donde N es el tamaño de muestra, yi es el valor real y ŷi es el valor estimado.

Raíz de error cuadrático medio (RMSE). La función de esta medida es realizar una diferencia entre los valores estimados y los valores reales, estas diferencias son elevadas al cuadrado y se calcula el promedio de todas ellas, posteriormente se saca la raíz cuadrada de este promedio.[34]

$$RMSE = \sqrt{MSE}$$

Coeficiente de determinación (R^2). El coeficiente de determinación es una de las métricas más usadas, indica que tan ajustada está la línea de regresión respecto a los resultados reales. La métrica es medida de 0 a 1, siendo 1 el valor más cercano y el indicador de que los datos se ajustan perfectamente[35]. Se define mediante la siguiente ecuación:

$$R^2 = 1 - (\frac{MSEp}{MSEm})$$

Donde MSEp es el error cuadrático medio de los valores estimados y el MSEm es el error cuadrático medio de los valores reales.

[33] NEGRÓN, Pablo Andrés. Redes Neuronales Sigmoidal Con Algoritmo Lm Para Pronostico De Tendencia Del Precio De Las Acciones Del IPSA. Pontificia Universidad Católica de Valparaíso. 2014.
[34] NEGRÓN. Op. Cit.
[35] Ibid.

2.3.5. Coeficiente de correlación de Pearson y Spearman. Los coeficientes de asociación son valores de orden cuantitativo que permiten saber qué tan ajustados linealmente se encuentran dos variables, entre estos coeficientes los más importantes y utilizados son el coeficiente de Pearson y Spearman.[36] A pesar de que ambos son coeficientes de asociación, existe una particular diferencia entre ellos, el coeficiente de Pearson es paramétrico, es decir, infiere los resultados respecto a la población real, lo que hace necesario que la distribución de la muestra se asemeje a la distribución real, por eso, solo se podrá hacer uso de variables cuantitativas que cumplan con el requisito de una distribución normal, por otro lado, la correlación de Spearman es no paramétrica, por lo que sus valores solo son representativos en los parámetros no poblacionales, mientras que su uso es más amplio ya que se ajusta a variables tanto cualitativas como cuantitativas.[37]

Tabla 4. Criterios de clasificación correlaciones de Pearson y Spearman

Valor	Criterio
r=1.00	Correlación grande, perfecta y positiva
0.90 <= r <1.00	Correlación muy alta
0.70<= r < 0.40	Correlación alta
0.40 <= r <0.70	Correlación moderada
0.20 <= r <0.40	Correlación muy baja
r = 0	Correlación nula
r=-1.00	Correlación grande, perfecta, negativa

Fuente: DIAZ, Ignacio, et al. Guía de asociación entre variables (Pearson y Spearman en SPSS). Universidad de Chile. Santiago de Chile. 2014.

En la tabla 4 se muestran los valores en lo que se mide un coeficiente de relación, siendo 1 la correlación perfecta positiva y -1 una correlación perfecta inversa o negativa, los valores más cercanos a 0 indican que la correlación es nula.

[36] DIAZ, Ignacio, et al. Guía de asociación entre variables (Pearson y Spearman en SPSS). Universidad de Chile. Santiago de Chile. 2014.
[37] DIAZ, Op. Cit.

2.4. PERFORACIÓN DE POZOS

La realización de este proyecto de grado está enfocada en la predicción del comportamiento de la variable ROP usando diferentes modelos de machine learning para una formación dura y abrasiva, sin embargo, para llegar a cumplir el objetivo es necesario el uso de diferentes parámetros que interfieren en el proceso como inputs para llegar a una predicción final. Estos parámetros están relacionados con el proceso de perforación de un pozo petrolero, por ello, en esta sección se definirán algunos conceptos claves dentro del proceso de perforación de pozos.

En el año 2020, Herbert[38] definió la perforación como una operación compleja de ingeniería, donde todos los sistemas y equipos utilizados se eligen de acuerdo al objetivo por el que se lleva a cabo, ya sea verificar la existencia de hidrocarburos, las cantidades y límites de los yacimientos o para la extracción del mismo, para ello se debe tener en cuenta la dureza de la formación o roca que se pretende perforar, el diámetro del pozo, la profundidad y las problemáticas que se afrontaran durante la operación.

Durante este proceso, existen diferentes factores que pueden aportar más o menos efectividad a la operación, uno de ellos y que compete en esta investigación es la dureza y abrasividad de la formación que se perfora. Aquellas formaciones que se pueden considerar abrasivas son aquellas que tienen la capacidad de desgastar la superficie de contacto de otro cuerpo más duro por medio de fricción o rozamiento, esto se debe a la presencia de diversos minerales dentro de la composición de la roca como pirita, pedernal, magnetita, entre otros, estos materiales contribuyen al desgaste excesivo y prematuro de la broca de perforación, siendo el calibre el parámetro más afectado.[39]

[38] HERRERA, Herbert. Ingeniería de perforación de pozos de petróleo y gas. Universidad Politécnica de Madrid. 2020.
[39] UNAM. Aspectos generales relacionados al corte de núcleos. Universidad Nacional Autónoma de México [Online]

Dentro de los factores que aumentan la capacidad abrasiva de la roca son: El tamaño de los granos donde generalmente los granos más grandes son más abrasivos que los más pequeños, el ordenamiento ya que esta propiedad agrupa los granos por tamaño, por lo tanto, en los depósitos donde los sedimentos no fueron ordenados y poseen granos de diferentes tamaños suelen ser más porosas y abrasivas que las que están bien ordenadas. Otro factor crucial es la forma de los granos, este es dependiente del tipo de erosión, las rocas son más redondas o angulosas dependiendo del proceso de transporte al cual fueron sometidas, las rocas más angulosas tienden a ser más abrasivas. La última propiedad influyente en estas características es la compactación y cementación, las formaciones que no se encuentran bien consolidadas suelen ser más abrasivas, al igual que por ejemplo una arenisca dura y bien cementada, en donde las tasas de perforación tienden a ser más bajas de lo normal.[40]

Tabla 5. Clasificación por dureza y abrasividad según el tipo de roca

	Dureza y abrasividad				
	Abrasiva dura	**Abrasiva menos dura**	**Abrasiva friable**	**Abrasiva menos friable**	**No abrasiva blanda**
Rocas	- Conglomerados con cuarzo. - Areniscas. - Grauvacas. - Ortocuarcitas.	- Ceniza volcánica. - Ceniza silícea. - Areniscas de grano grueso. - Tobas.	- Areniscas friables. - Areniscas calcáreas. - Gravas consolidadas.	- Calizas - Arcillas esquistosas - Cretas	- Margas. - Lutitas. - Carbones. - Yesos. - Calizas oolíticas. - Evaporitas.

Fuente: UNAM. Aspectos generales relacionados al corte de núcleos. Universidad Nacional Autónoma de México [Online].

[40] UNAM. Op. Cit.

En la tabla 5 se muestra una clasificación de los tipos de roca respecto a su dureza y abrasividad, siendo los conglomerados de cuarzo, las areniscas, grauvacas y ortocuarcitas las más abrasivas y duras. Cabe destacar que los datos utilizados en este proyecto fueron obtenidos de una formación catalogada como de alta dureza y abrasividad, razón por la cual los valores de tasa de ROP son bajos debido a los diferentes problemas que suelen presentarse en este tipo de operaciones.

2.4.1. Parámetros de perforación. A través de todo el proceso de perforación, se obtienen diferentes parámetros y mediciones que se realizan en tiempo real por medio de sensores, donde se mide segundo a segundo o en ocasiones en intervalos más grandes de tiempo, diferentes registros como profundidad, ROP, WOB, torque, entre otros. Esta investigación tiene como objetivo implementar algunos parámetros para poder realizar una predicción de la tasa de perforación. La tabla 6 muestra las unidades de medida, siglas y una pequeña definición de cada uno de los parámetros del dataset inicial de los datos recolectados.

Tabla 6. Lista de parámetros de perforación

Parámetro	Unidades	Descripción
ROP	ft/hr	Velocidad a la que se profundiza durante la perforación.
BIT_DEPTH	ft	Posición de la broca en profundidad.
GEN_TIME_DEPTH	hr	Profundidad de perforación que se genera en el tiempo.
HKLD	Klb-f	Peso ejercido sobre el gancho.
WOB	Klb-f	Peso ejercido sobre la broca.
TORQUE	kft.lb	Fuerza creada por la sarta de perforación debido a la rotación.
BIT_RPM	RPM	Velocidad a la que la broca rota durante la operación.
MOTOR_RPM	RPM	Velocidad de rotación del motor.

SURF_RPM	RPM	Velocidad de rotación en superficie.
PUMP	PSI	Presión de circulación del fluido de perforación circulando en la tubería.
FLOW_IN	USgal/min	Cantidad de fluido de perforación que ingresa en el sistema.
FLOW_OUT	%	Cantidad de fluido de perforación que sale del sistema medido en porcentaje.
OVERBALANCE	ppg	Diferencia entre la densidad del fluido de perforación y la densidad equivalente de presión de poro.
ΔTEMPERATURE	F°	Diferencia entre la temperatura de entrada y salida del pozo.
ROT_TIME	hr	Tiempo durante el cual la sarta de perforación se encuentra en rotación acumulada.
ON_BOTTOM_TIME	hr	Tiempo durante el cual la sarta se encuentra en fondo.
CUM_CIRC_TIME	hr	Tiempo acumulado durante el que circula lodo a través del sistema.
DESGASTE	Adimensional	Desgaste con que sale la broca después de la corrida.

Fuente: Autores

3. METODOLOGÍA Y DESARROLLO DEL PROYECTO

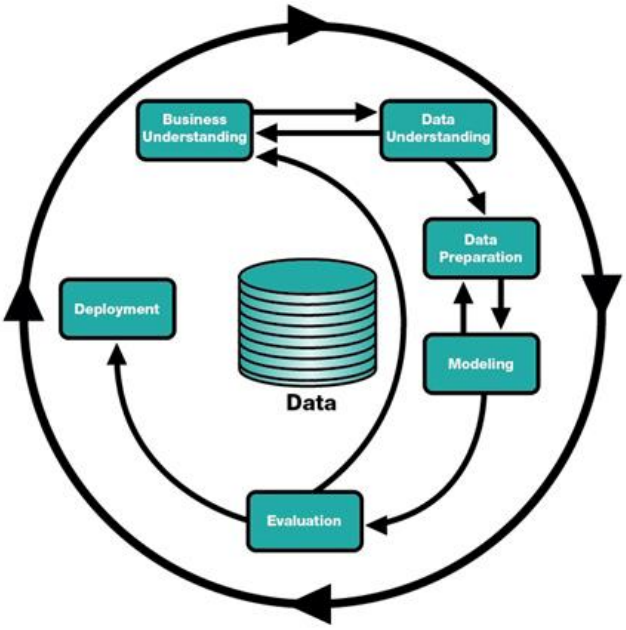
En esta sección se expone la metodología propuesta para desarrollar la presente investigación con el modelo de referencia CRISP-DM (Cross Industry Standard Process for Data Mining) que proporciona una descripción del ciclo de vida de un proyecto estándar de análisis de datos con el objetivo de estandarizar y normalizar el trabajo desarrollado en minería de datos de forma análoga a como se hace en la ingeniería del software con los modelos de ciclo de vida de desarrollo de software.

Fayyad[41] considera la minería de datos (DM) como una de las fases del proceso. El ciclo de vida de minería de datos consiste en seis fases principales (figura 6). La secuencia de las fases no es totalmente rígida, permite cierta flexibilidad, moverse atrás y adelante entre las diferentes fases siempre es requerido. Todas las fases de la metodología son consecuentes, es decir, las tareas o actividades que se realicen en una fase previa van a acondicionar las siguientes. Las flechas indican las dependencias entre fases que son más importantes y frecuentes.

El círculo exterior simboliza la naturaleza cíclica misma de la minería de datos, hay que tener en cuenta que el proceso de minería de datos no queda atrás cuando una solución es desplegada del ciclo y puesta en práctica. El aprendizaje obtenido durante el proceso y la solución desplegada puede desencadenar nuevas y más enfocadas preguntas relacionadas con el negocio.

[41] FAYYAD, Usama. et al. Knowledge Discovery and data mining: Towards an Unifying framework. 1996.

Figura 6. Modelo CRISP-DM



Fuente: SHAFIQUE, Umair y QAISER, Haseeb. A Comparative Study of Data Mining Process Models (KDD, CRISP-DM and SEMMA). University of Gujrat. 2014.

3.1. ENTENDIMIENTO DEL NEGOCIO

Business understanding o entendimiento del negocio, se posiciona como la primera y quizás una de las secciones más importantes dentro de un proyecto de ciencia de datos, en este se adicionan las tareas como comprensión de objetivos, requisitos del proyecto desde un punto de vista del negocio, objetivos técnicos y un plan de proyecto. Esta sección se hace fundamental ya que un proyecto sin una estructuración y objetivos bien definidos por sí solo no puede lograr tener resultados satisfactorios por más sofisticado que pueda ser el algoritmo.[42] A través del proyecto se presentaron diferentes aspectos fundamentales como antecedentes y

[42] GALAN CORTINA, Victor. Aplicación De La Metodología Crisp-Dm A Un Proyecto De Minería De Datos En El Entorno Universitario. Universidad Carlos III de Madrid. 2015.

ambientación petrolera para comprender la problemática y razón de esta investigación y cómo la minería de datos se convirtió en una solución para el mismo.

Durante la perforación de un pozo en la industria petrolera, es fundamental conocer el desempeño de parámetros como la ROP (Rate of Penetration) para poder determinar cómo llegar a un óptimo rendimiento y a su vez poder disminuir tiempos y costos de perforación, ya que una estimación precisa de la ROP puede beneficiar la planificación del pozo y evitar problemas operacionales inesperados. La mayoría de los modelos que han explorado la predicción de la ROP se basan en algoritmos empíricos en donde solo se consideraban parámetros de perforación limitados, estos modelos podrían perder fácilmente su rendimiento en entornos de perforación complicados como en formaciones duras y abrasivas.

La analítica descriptiva permite saber qué pasa en campo, por ejemplo ¿qué ROP se tiene por formación?, ¿qué ocurre con esos datos?, o si se tiene una ROP alta o baja, este análisis estadístico se puede realizar con diferentes herramientas informáticas de análisis de datos, pero si se quiere ir al siguiente paso, añadiendo más valor, naturalmente con más dificultad, se debe cuestionar ¿por qué pasa esto?, implementando el análisis diagnóstico, estadísticas descriptivas y diferenciales, que llevan a las primeras conclusiones acerca de los datos.

A partir de allí se exhibe el proceso de predicción de la ROP, para poder dejar a un lado las suposiciones y comenzar a leer los datos, entenderlos y procesarlos mostrando relaciones estadísticas para entender no solo qué pasa en el pozo o el por qué se obtuvo determinada ROP sino también poder responder ¿qué pasará?, ¿qué ROP se obtendrá para determinada formación? donde no solo se evalúe la situación, proceso o nivel de eficiencia, sino también sacar provecho de lo que podría suceder en el futuro con esta variable mediante el uso de machine learning.

Con base en lo anterior, la metodología propuesta para desarrollar la presente investigación inicia con la depuración de la base de datos obtenidos de una fuente real pero debido a razones de confidencialidad no se mencionan los nombres de los

pozos y datos de la formación de estudio. Esta base de datos está conformada por los parámetros de perforación de cuatro pozos, comenzando con la asignación de topes de la formación objeto de estudio, filtrado del dataset según el tiempo de perforación, dejando así solo los datos del tiempo efectivo y, por último, detección y tratamiento de valores faltantes sí los hay. A partir de allí, se realiza un análisis exploratorio de datos EDA, el cual busca obtener una descripción estadística de las variables, permitiendo identificar y eliminar valores atípicos de la información con el fin de generar una base de datos consistente. Luego se realiza el cálculo y creación de nuevas variables como diferencia de temperatura y desgaste de la broca.

Llegados a este punto se procede a usar la biblioteca de pycaret del lenguaje de programación Python, que con la función compare_models permite determinar los mejores modelos de aprendizaje automático supervisado que se ajustan al dataset y sus respectivas métricas, procediendo a seleccionar los más adecuados y realizar su desarrollo con la biblioteca de software de aprendizaje automático, Scikit learn, con las proporciones 60/30/10 del dataset, para su entrenamiento, prueba y validación respectivamente. Después se realiza la optimización o tuneo de los modelos con el marco de software de ajuste de hiperparámetros automático Optuna permitiendo obtener un coeficiente de determinación (r2) de al menos 70% en los modelos de aprendizaje.

Finalmente, se desarrolla un microservicio para la predicción de la ROP con la biblioteca de streamlit para implementar un aplicativo web en la plataforma como servicio (PaaS) Heroku, usando el modelo con la mejor métrica y que además cumpla con la capacidad máxima que permite la plataforma (500 MB) con el fin de permitir al usuario saber qué ROP se obtendrá en la formación objeto de estudio en las siguientes corridas. En la tabla 7, se presentan las librerías de python usadas en el procesamiento de los datos a lo largo del proyecto y en sus diferentes etapas.

Tabla 7. Lista de librerías de python utilizadas.

Librerías	Descripción
Pandas	Pandas es una de las librerías de python más útiles para los científicos de datos y se destaca por lo fácil y flexible que hace la manipulación de datos y el análisis de datos. Las estructuras de datos principales en pandas son Series para datos en una dimensión y DataFrame para datos en dos dimensiones.
Numpy	NumPy proporciona una estructura de datos universal que posibilita el análisis de datos y el intercambio de datos entre distintos algoritmos. Las estructuras de datos que implementa son vectores multidimensionales y matrices con capacidad para gran cantidad de datos.
Matplotlib	Matplotlib es la librería gráfica estándar de python y la más conocida. Se puede usar para generar gráficos de calidad necesaria para publicarlas tanto en papel como digitalmente.
Scikit learn	Scikit-learn es una librería de python para Machine Learning y Análisis de Datos. Está basada en NumPy, SciPy y Matplotlib. Las ventajas principales de scikit-learn son su facilidad de uso y la gran cantidad de técnicas de aprendizaje automático que implementa.
Seaborn	Seaborn es una librería gráfica basada en matplotlib, especializada en la visualización de datos estadísticos. Se caracteriza por ofrecer un interfaz de alto nivel para crear gráficos estadísticos visualmente atractivos e informativos.
Pycaret	PyCaret es una biblioteca de aprendizaje automático de código bajo y de código abierto en Python que automatiza los flujos de trabajo de aprendizaje automático. Es una herramienta de gestión de modelos y aprendizaje automático de extremo a extremo que acelera el ciclo del experimento de manera exponencial y lote hace más productivo.
Plotly	Plotly es una plataforma web colaborativa para la visualización y el análisis de datos. Crea gráficos interactivos con calidad de publicación.
Joblib	Joblib es un conjunto de herramientas para proporcionar canalización ligera en Python. Está optimizado para ser rápido y robusto en datos grandes en particular y tiene optimizaciones específicas para matrices numerosas.
Optuna	Es un marco de optimización de hiperparámetros de código abierto para automatizar la búsqueda de hiperparámetros.
Lasio	Lasio es un paquete de Python 3 para leer y escribir archivos de registro estándar ASCII (LAS), que se utiliza para datos de pozo como registros geofísicos, geológicos o petrofísicos.

Statistics	Este módulo proporciona funciones para calcular estadísticas matemáticas de datos numéricos.
Pickle	Este módulo nos permite almacenar fácilmente colecciones y objetos en ficheros binarios abstrayendo toda la parte de escritura y lectura binaria.
Streamlit	Streamlit es una biblioteca de Python de código abierto que facilita la creación y el intercambio de hermosas aplicaciones web personalizadas para el aprendizaje automático y la ciencia de datos.

Fuente: DOWNEY, A.B. How to Think Like a Computer Scientist. Learning with Python. Green Tea Press. 2008.

3.2. ANÁLISIS EXPLORATORIO Y DESCRIPTIVO

El análisis exploratorio y descriptivo se conoce como Business Intelligence, en esta etapa se capturan los datos que a su vez son transformados en datos más depurados y elaborados, con mayor utilidad, buscando entender el nivel de correlación de la data y con el propósito de explicar las situaciones que se han presentado. Se responde aquí a la pregunta qué sucede, qué dicen los datos acerca del comportamiento de los parámetros de perforación y el impacto de éstos en la predicción de la ROP. En la siguiente sección se realiza la preparación de la base de datos con el fin de generar una dataset consistente para implementar en el modelo predictivo.

3.2.1. Comprensión de los datos. Continuando con la metodología planteada por CRISP-DM, el Data understanding o entendimiento de los datos es la segunda fase de este, donde su objetivo principal es comprender la relación principal entre el objetivo del problema y los datos que se tienen, se espera que exista una identificación de objetivos y familiarización con la base de datos inicial. Esta fase, junto con las siguientes dos, son las que demandan mayor tiempo y esfuerzo dentro

de un proyecto de minería de datos.[43] Esta sección será complementada con fases posteriores donde se dará una mayor ambientación a los datos y a la manera en que estos se relacionan entre sí.

Se adquieren datos dinámicos resultantes de diferentes corridas durante la perforación de cuatro pozos petroleros (pozo A, B, C y D) para una formación específica caracterizada por ser dura y abrasiva. Estos datos compilan la información de perforación en series de tiempo cada 10 segundos, con la información de 18 variables numéricas como la ROP y parámetros como profundidad, RPM, torque, WOB, hook load, caudal, presión de la bomba, temperatura del lodo, tiempo de rotación, entre otras (Tabla 8), que son afectados por las características de la formación objeto de estudio.

Tabla 8. Variables de los datos recolectados de cuatro pozos perforados en la formación objeto de estudio

Variables	Unidades	Tipo de variable
ROP	ft/hr	Numérica
Bit Depth	ft	Numérica
Gen-time depth	ft	Numérica
Hook load	Klb-f	Numérica
WOB	Klb-f	Numérica
Torque	Kft.lb	Numérica
Bit RPM	RPM	Numérica
Motor RPM	RPM	Numérica
Surf RPM	RPM	Numérica
Pump	PSI	Numérica
Flow out PC	%	Numérica
Flow in	USgal/min	Numérica
Overbalance	ppg	Numérica
Mud Temperature out	°F	Numérica
Mud temperature in	°F	Numérica

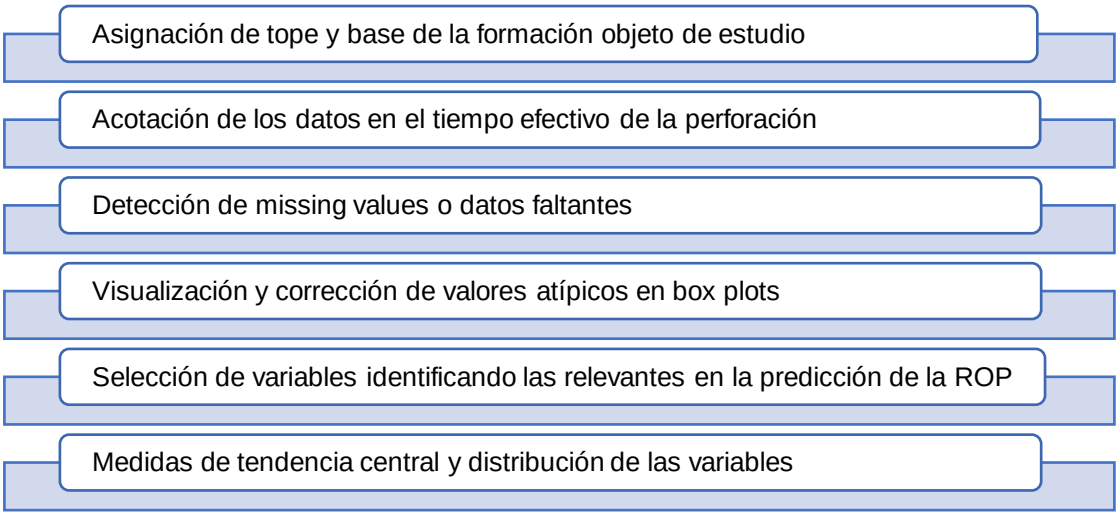
[43] GALAN CORTINA. Op. Cit.

Rot time	hr	Numérica
On bottom time	hr	Numérica
Cum circ time	hr	Numérica

Fuente: Autores.

3.2.2. Preparación de los datos. Las actividades necesarias para construir el conjunto de datos final que servirá para alimentar el modelo (figura 7) en etapas próximas están organizadas así:

Figura 7. Actividades designadas para la preparación de los datos



3.2.2.1. Asignación de tope y base de la formación objeto de estudio. El dataset está conformado por 18 variables y 7’184.103 observaciones de los datos de perforación recolectados de 4 pozos. De los cuales, se extrae solo la información de interés, es decir, los datos de la formación objeto de estudio, por lo tanto, se realiza la acotación por tope y base mostrada en la tabla 9.

Tabla 9. Cantidad de datos recolectados de cada pozo

	Datos iniciales
Pozo A	2,677,681
Pozo B	2,984,023
Pozo C	894,138
Pozo D	628,261
Total	7,184,103

El pozo A fue perforado hasta los 20808 ft, el pozo B hasta los 19100 ft, el pozo C hasta los 18320 ft y el pozo D hasta los 19440 ft, pero las profundidades donde se encuentra la formación objeto de estudio son las siguientes (Tablas 10, 11, 12 y 13):

Tabla 10. Tope y base de la profundidad de la formación objeto de estudio, en el pozo A

	POZO A	
Formación de estudio	**Tope (ft)**	17220
	Base (ft)	17712
	Tope (ft)	18033
	Base (ft)	18441

Tabla 11. Tope y base de la profundidad de la formación objeto de estudio, en el pozo B

	POZO B	
Formación de estudio	**Tope (ft)**	17664
	Base (ft)	18488

Tabla 12. Tope y base de la profundidad de la formación objeto de estudio, en el pozo C

	POZO C	
Formación de estudio	**Tope (ft)**	16195
	Base (ft)	16911

Tabla 13. Tope y base de la profundidad de la formación objeto de estudio, en el pozo D

	POZO D	
Formación de estudio	**Tope (ft)**	16189
	Base (ft)	16721
	Tope (ft)	17371
	Base (ft)	17937

Al acotar los datos por tope y base, se elimina el 86.5% de los datos del pozo A, quedando 361.600 datos de 2'677.681, del pozo B se elimina el 73.39% quedando 793.922 datos de 2'984.023, del pozo C se elimina el 93.30% de los datos quedando 59.912 datos de 894.138 y del pozo D se elimina el 80.54% quedando 122250 datos de 628.261, como se puede observar en las tablas 14 y 15.

3.2.2.2. Acotación de los datos en el tiempo efectivo en la perforación. El tiempo de operación se divide entre el tiempo efectivo, es decir el tiempo en el que la broca estuvo en fondo perforando y el tiempo de viaje donde no hay avance o cambio en la profundidad, en este caso, para el estudio de la ROP se necesitan los datos del tiempo efectivo, por lo tanto, se eliminan los datos donde no haya cambio de profundidad, es decir los valores duplicados de profundidad generada en el tiempo.

En el pozo A se eliminan 317122 datos duplicados de profundidad generada en el tiempo que corresponde al 87.7% de los datos de la formación objeto de estudio, quedando un total de 44478 datos en este pozo. En el pozo B se eliminan 747563 datos, que corresponde al 94.16% de los datos de la formación objeto de estudio, quedando un total de 46359 datos en este pozo. En el pozo C se eliminan 37021 datos, que corresponde al 61.79% quedando un total de 22891 datos en este pozo. Y en el pozo D se eliminan 92393 datos, que corresponde al 75.58% de los datos de la formación objeto de estudio quedando un total de 29857 datos en este pozo. (Tablas 14 y 15).

Tabla 14. Número de datos restante después de filtrar los datos de la formación objeto de estudio y los datos de tiempo efectivo en la perforación

Número de datos restantes		
Pozos	**Acotación por tope y base**	**Acotación por tiempo efectivo de perforación**
A	361,600	44,478
B	793,922	46,359
C	59,912	22,891
D	122,250	29,857
TOTAL	**1,337,684**	**143,585**

Tabla 15. Porcentaje de datos eliminados después de filtrar los datos de la formación objeto de estudio y los datos de tiempo efectivo en la perforación

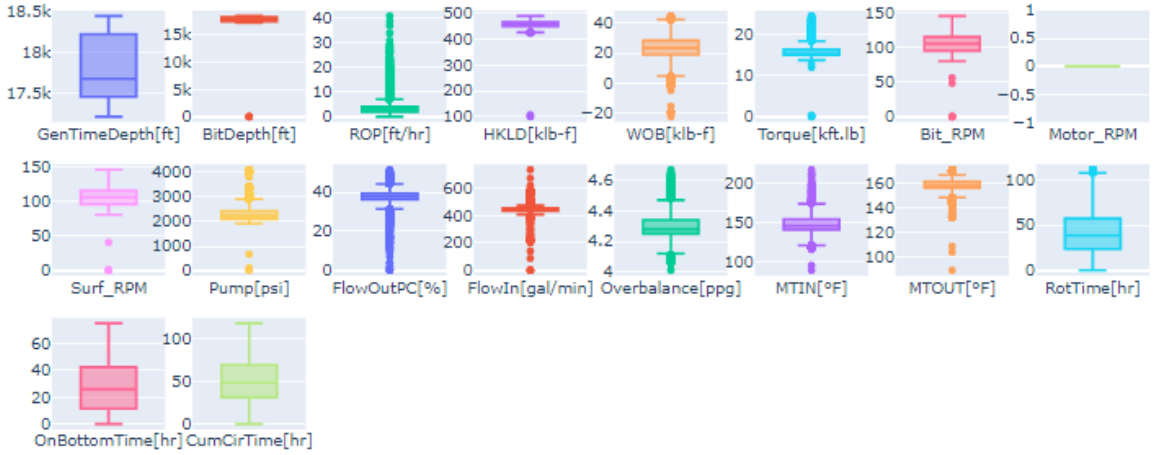
Porcentajes eliminados		
Pozos	**Acotación por tope y base**	**Acotación por tiempo efectivo de perforación**
A	86.50%	87.70%
B	73.39%	94.16%
C	93.30%	61.79%
D	80.54%	75.58%

3.2.2.3. Detección de missing values o datos faltantes. Los datos faltantes o missing values se definen como valores no disponibles que serían útiles o significativos para el análisis de los resultados. Con el uso del lenguaje de programación de Python se crea la función missing_values para contar la cantidad de datos faltantes en el dataset, esta función permite evidenciar que la base de datos no cuenta con missing values o valores faltantes.

3.2.2.4. Visualización y corrección de valores atípicos en Box plots. En esta subfase se genera el diagrama de caja o box plot, representación visual importante para la preparación y limpieza de los datos, que describe al mismo tiempo varias características importantes de datos, tales como el centro, la dispersión, la

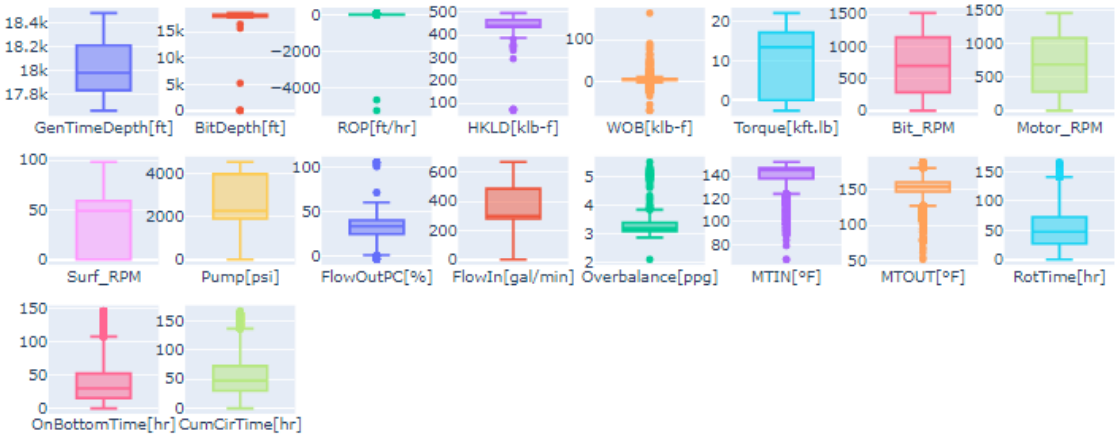
desviación de simetría y la identificación de observaciones que se alejan de manera poco usual del resto de los datos, conocidos como valores atípicos.

Figura 8. Box plots o diagramas de caja de las variables numéricas del pozo A



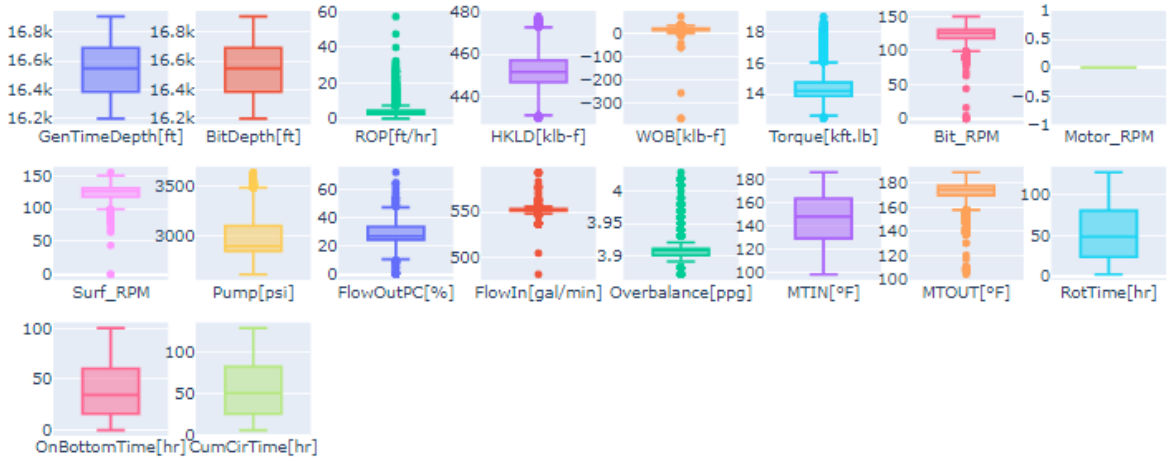
Fuente: Autores.

Figura 9. Box plots o diagramas de caja de las variables numéricas del pozo B



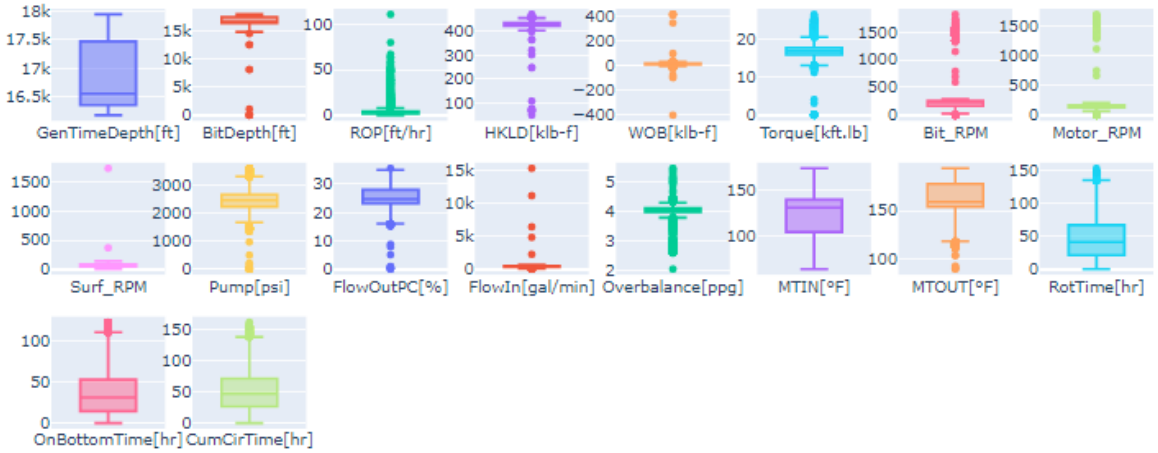
Fuente: Autores.

Figura 10. Box plots o diagramas de caja de las variables numéricas del pozo C



Fuente: Autores.

Figura 11. Box plots o diagramas de caja de las variables numéricas del pozo D



Fuente: Autores.

Como se puede observar en la figura 8 de los diagramas de caja de cada variable numérica de los datos del pozo A, se encuentran valores atípicos en los parámetros: Bit_depth, HKLD, WOB, Torque, Bit_RPM, Surf_RPM y Pump. En la figura 9 se

encuentran valores atípicos en los parámetros: Bit_depth, ROP, HKLD, Flow out y WOB. En los diagramas de caja de los datos del pozo C de la figura 10, se puede observar que se encuentran valores negativos en el parámetro WOB. Y en los diagramas de caja de los datos del pozo D (figura 11), se encuentran valores atípicos en los parámetros: Bit_depth, WOB, Torque, Surf_RPM y Flow_in.

En consecuencia, se acotan los datos eliminando los valores de bit depth que se salgan del rango de la profundidad de la formación objeto de estudio con respecto al tope y base indicados en las tablas 10, 11, 12 y 13. Se eliminan también los valores atípicos que se encuentran en la distribución de datos de ROP, hook load, flow out torque, RPM de la broca, RPM en superficie y en los datos de presión de la bomba. Y finalmente, se eliminan los valores negativos e iguales a cero que se encuentran en la distribución de datos de surf RPM, WOB y torque.

Después de realizar la limpieza de datos eliminando los outliers o valores atípicos para conseguir crear una base de datos compacta, se presentan los datos restantes en las tablas 16 y 17.

Tabla 16. Número de datos restantes en cada pozo después de eliminar outliers

Pozos	Número de datos restantes
A	43,873
B	28,493
C	22,238
D	29,525
TOTAL	124,129

Tabla 17. Porcentajes de datos eliminados en cada pozo después de eliminar outliers

Pozos	Porcentajes eliminados
A	1.36%
B	38.54%
C	2.85%
D	1.11%

3.2.2.5. Selección de variables. La maldición de la dimensionalidad o maldición de la dimensión es un problema que se puede llegar a presentar si se tienen en cuenta todos los atributos posibles en un sistema. El desempeño de la predicción en machine learning depende de la cantidad de los datos y la calidad, es decir, a tener en consideración sólo aquellos atributos que realmente aporten datos de valor al modelo de aprendizaje automático. Considerar pensar que "entre más datos mejor" es un error que desembocará en un tiempo de procesamiento alto y en una selección de datos de entrada que son irrelevantes o redundantes para el sistema de regresión.[44]

Teniendo en cuenta lo anterior, es importante seleccionar solo las variables relevantes para la predicción de la ROP. Parámetros como temperaturas del lodo, tiempo acumulado, en fondo y de rotación, además de la variable de profundidad generada en el tiempo pueden ser variables redundantes que afecten el rendimiento del modelo de aprendizaje.

En consecuencia, se analiza qué variables de cada uno de los tipos mencionados se descartan o se dejan en el dataset. En el caso de las profundidades se elige trabajar con la profundidad de la broca (bit depth) y se elimina la variable Gen-Time depth ya que tienen los mismos datos y sería redundante usar las dos variables. De igual forma ocurre con las tres variables de tiempo, tiempo de fondo (on bottom time), tiempo de circulación acumulado (cum circ time) y tiempo de rotación (rot time), se considera que las variables on bottom time y cum circ time no son relevantes para la predicción de la ROP, entonces se deja solamente la variable de tiempo de rotación.

Debido a que las RPM de la broca (bit RPM) son equivalentes a la suma de las RPM en superficie (surf RPM) más las RPM del motor (motor RPM), se decide dejar solamente bit RPM. Finalmente, con respecto a las variables de temperatura del lodo, se añade la variable de diferencia de temperatura (dT), ya que se conoce la

[44] AGUILAR, Adán. Un algoritmo inspirado en la naturaleza para resolver problemas de optimización numérica restringida con alta dimensionalidad. Universidad Veracruzana. 2019.

temperatura de entrada y de salida del lodo, creando la nueva variable y eliminando las variables de mT in y mT out.

Desgaste de la broca. El desgaste de la broca es un parámetro importante en la perforación de pozos debido a que dependiendo del grado de dureza y abrasividad de la roca este puede generar un efecto adverso en el avance de la perforación, haciendo que la broca no tenga el mismo rendimiento y la ROP se vea afectada. Con los datos obtenidos del bit record de la perforación de cada pozo se puede conocer el desgaste que tiene la broca después de cada corrida. A pesar de que se cuenta con el valor del desgaste al final de la corrida, se desconoce la tendencia que tiene el desgaste en el intervalo. Debido a esto, suponemos un desgaste lineal y un desgaste radical para observar la correlación de cada uno con la ROP.

En el pozo A se realizaron 8 corridas en la formación objeto de estudio, en el pozo B 6 corridas, en el pozo C 3 corridas y en el pozo D 8 corridas, pero ya que se desconoce el desgaste de las demás formaciones de los pozos perforados se deben eliminar los datos de la broca que viene de otra formación. Teniendo en cuenta esto, se eliminaron 4034 datos (9.19%) del pozo A, 988 datos (3.47%) del pozo B, 2412 datos (10.852%) del pozo C y 1095 datos (3.71%) del pozo D. Dejando como resultado un dataset de 115600 observaciones (tablas 18 y 19).

Tabla 18. Número de datos restante después de eliminar los datos de las primeras brocas de cada corrida

Pozos	Número de datos restantes
A	39,839
B	27,505
C	19,826
D	28,430
TOTAL	115,600

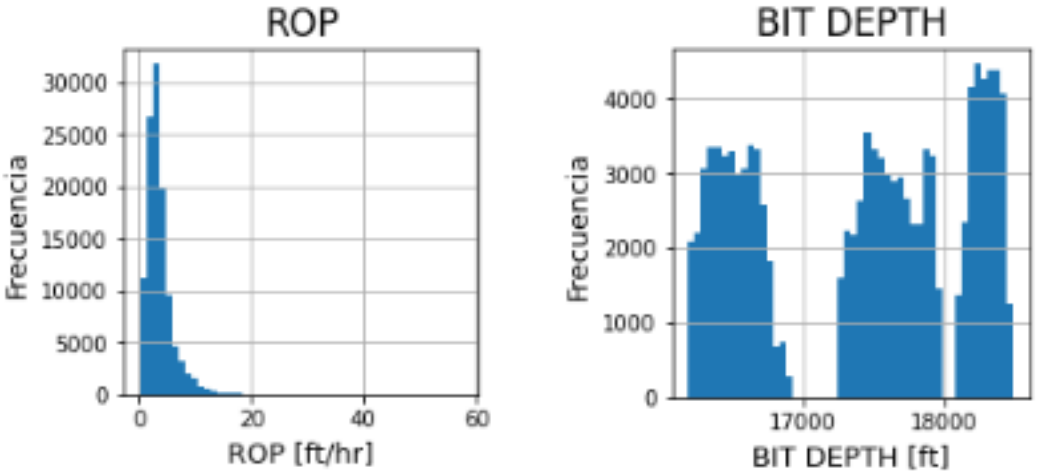
Tabla 19. Porcentaje de datos eliminados de las primeras brocas de cada corrida

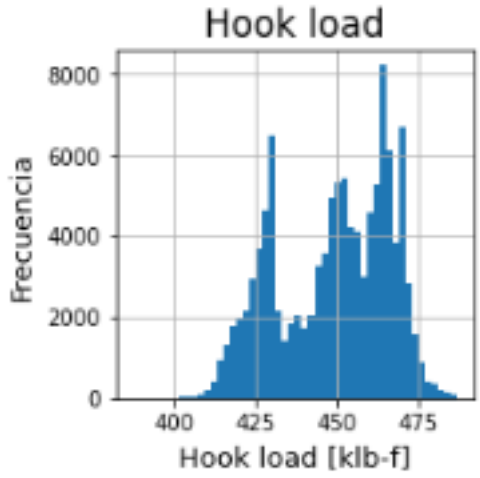
Pozos	Porcentajes eliminados
A	9.19%
B	3.47%
C	10.85%
D	3.71%

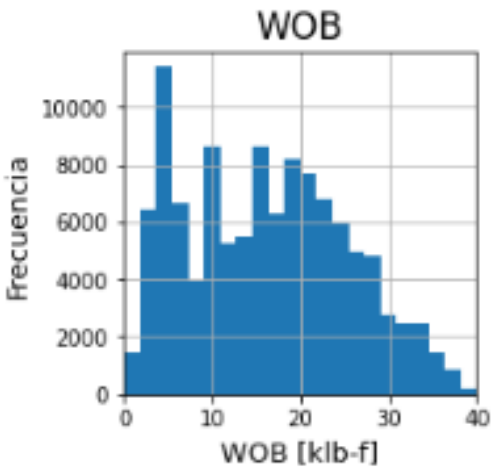
El dataset final queda conformado por las siguientes variables numéricas: Bit depth, ROP, hook load, WOB, torque, bit RPM, pump, flow out, flow in, overbalance, dT, rot time, desgaste lineal y desgaste radical.

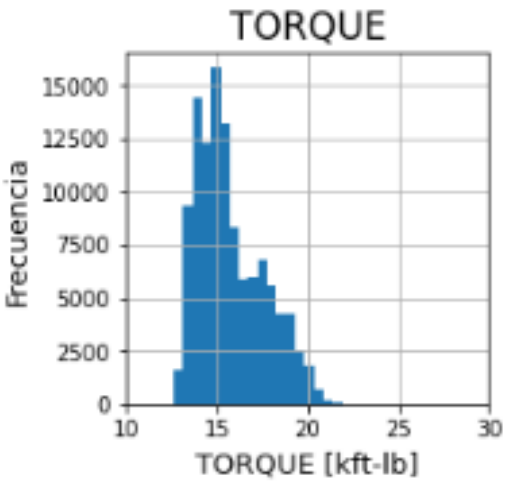
3.2.2.6. Medidas de tendencia central y dispersión. En estadística, la frecuencia (o frecuencia absoluta) de un evento es el número de veces en que dicho evento se repite durante un experimento o muestra estadística. Comúnmente, la distribución de la frecuencia suele visualizarse con el uso de histogramas.

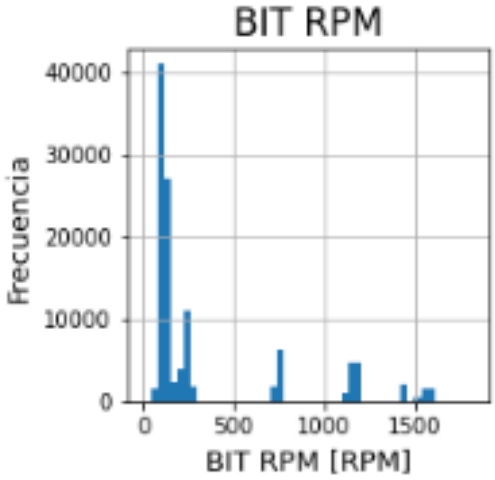
Figura 12. Histogramas de las variables numéricas

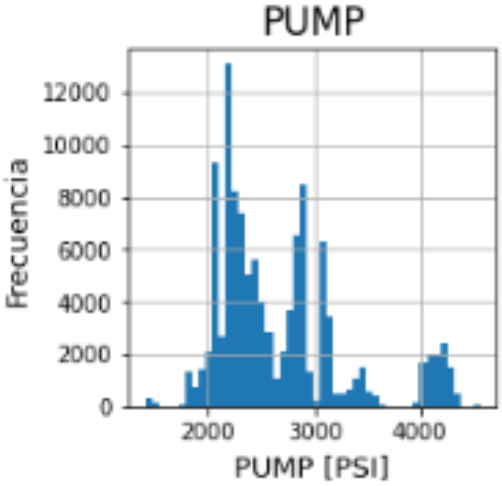


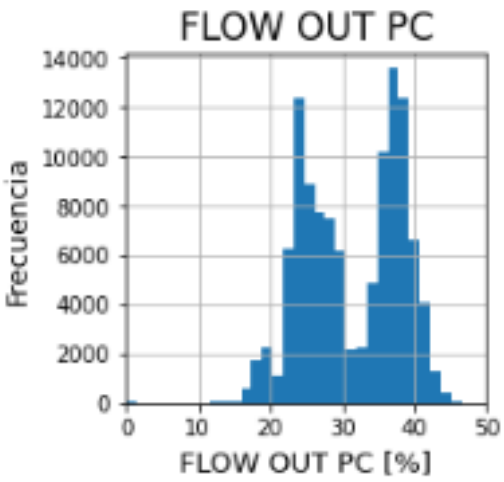

Hook load
Frecuencia
Hook load [klb-f]

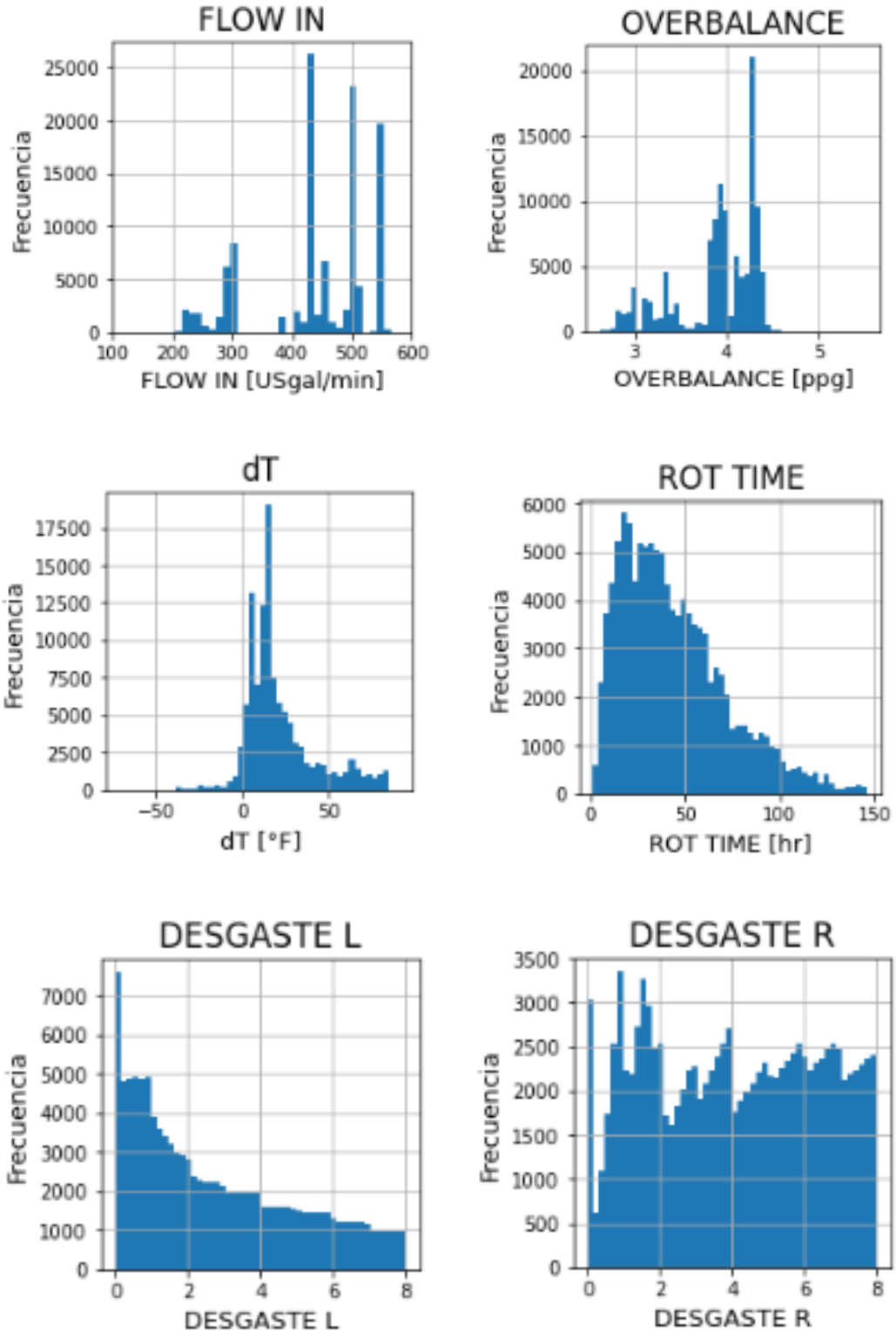

WOB
Frecuencia
WOB [klb-f]


TORQUE
Frecuencia
TORQUE [kft-lb]


BIT RPM
Frecuencia
BIT RPM [RPM]


PUMP
Frecuencia
PUMP [PSI]


FLOW OUT PC
Frecuencia
FLOW OUT PC [%]



Fuente: Autores.

En la figura 12 se pueden observar los histogramas de cada variable de entrada para la predicción y la variable dependiente ROP, en el caso de esta, los rangos para las secciones de la formación en estudio pueden oscilar entre 0 y 57.6 ft/hr, pero la mayor concentración de sus valores está entre los rangos 0 a 10 ft/hr, debido a que en la formación objeto de estudio la mayoría de las ROP obtenidas no son tan altas, afectando la eficiencia de la operación.

Los histogramas de las variables ROP, Torque y Rot_time presentan asimetría hacía la derecha. El torque concentra la mayor cantidad de valores en el rango de 14.24 a 17.05 Kft.lb. La variable Bit_depth presenta tres agrupaciones donde se concentran los datos en el histograma debido a que está filtrado por topes de la formación objeto de estudio en los cuatro pozos. El peso sobre la broca WOB concentra la mayor cantidad de valores en el rango 7.88 a 23 klbf. La mayor concentración de los datos de Hook load oscilan entre 433.23 y 464.27 klbf.

Con respecto al desgaste con tendencia lineal y con tendencia radical, la mayoría de los datos oscilan entre los rangos 0.85 a 4.4 y 1.89 a 6.04, respectivamente. El rango intercuartílico del desgaste radical es mayor por lo tanto sus datos están más dispersos. La variable revoluciones por minuto en la broca concentra la mayoría de sus datos entre el rango 100 a 251 RPM.

El caudal de entrada flow in concentra la mayor cantidad de valores en el rango que oscila entre 424 y 503 GPM y el porcentaje de flujo de salida flow out pc tiene la mayoría de los datos concentrados entre 27.46% y 37.55%. La mayor parte de los datos de la presión de la bomba Pump están entre 2207 y 2898 psi. El overbalance oscila entre los valores 2.62 y 5.53 psi pero la mayor cantidad de sus datos se concentran entre 3.8 y 4.26 psi. Finalmente, la variable de diferencia de temperatura presenta mayor asimetría hacía la izquierda y sus datos se consolidan más entre el rango de 8 a 27.6 grados Fahrenheit.

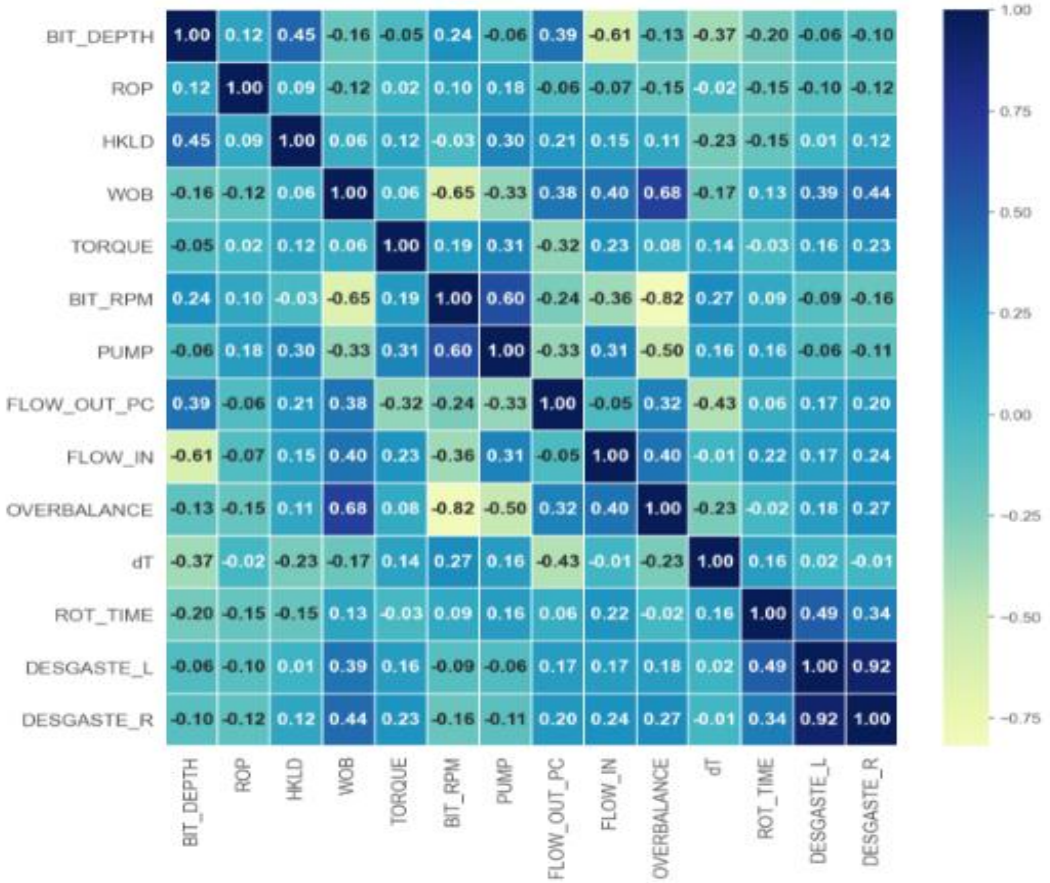
La tabla de medidas de tendencia central de los parámetros donde se presentan datos como el cuartil 1, cuartil 2, cuartil 3, el promedio de cada variable, la desviación

estándar, los valores mínimo y máximo y el promedio se puede encontrar en el anexo 1.

3.3. ANÁLISIS DIAGNÓSTICO

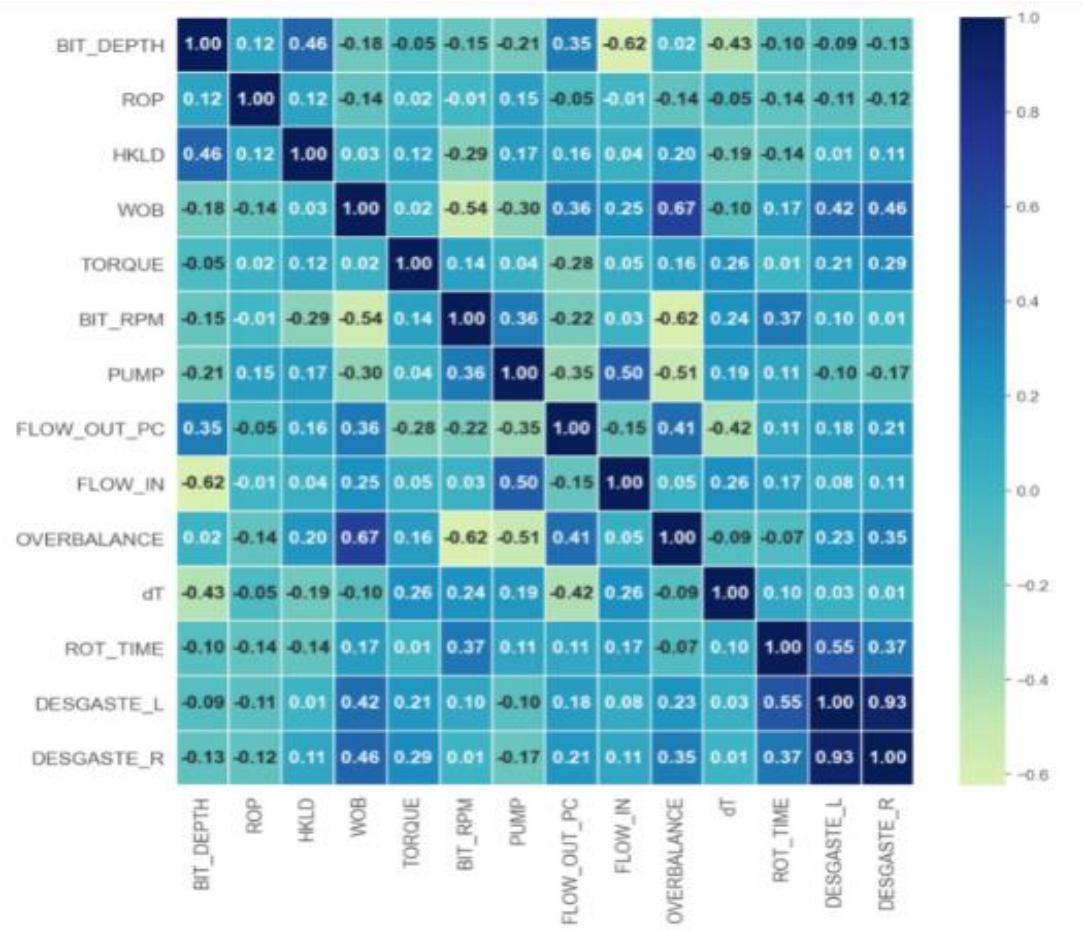
Análisis de correlación entre las variables. El coeficiente de correlación de Pearson es una medida de dependencia lineal entre dos variables aleatorias cuantitativas. El coeficiente de correlación de Spearman es una medida no paramétrica de la correlación de rango que mide la fuerza y la dirección de la asociación entre dos variables clasificadas. En las figuras 13 y 14 se presentan las matrices de correlación de Pearson y Spearman.

Figura 13. Matriz de correlación de Pearson



Fuente: Autores.

Figura 14. Matriz de correlación de Spearman



Fuente: Autores.

Las matrices permiten evidenciar el grado de correlación (directa con colores más oscuros e inversa con colores más claros) entre las variables de perforación. Se puede observar que no hay una correlación lineal de la ROP con las demás variables, pero las demás variables si se correlacionan entre ellas.

Los valores de la variable overbalance se ven afectados directamente por el peso sobre la broca con una correlación moderada y además tiene una correlación inversa alta con las revoluciones por minuto ya que al disminuir las RPM o al

aumentar el peso sobre la broca aumenta el overbalance en la perforación. De igual forma al aumentar el peso sobre la broca disminuyen las RPM de la broca

Con respecto al caudal, esta variable tiene una correlación inversa moderada con la profundidad de la broca. El desgaste con tendencia lineal y el desgaste con tendencia radical tienen una correlación directa con el peso sobre la broca, ya que al aumentar el peso aumenta el desgaste de la broca. La variable de desgaste con tendencia lineal tiene una correlación directa moderada con el tiempo de rotación, al aumentar el tiempo de rotación aumenta el desgaste de la broca. Finalmente, la presión de la bomba tiene una correlación directa moderada con las revoluciones por minuto y el caudal.

Es importante saber que con la implementación de los modelos de machine learning se va a analizar qué variables impactan más o tienen correlaciones no lineales con la ROP.

3.4. ANÁLISIS PREDICTIVO

Una vez que se ha culminado la fase de preparación de los datos y se cuenta con un dataset consistente de 14 variables (tabla 20), se procede a implementar los modelos de aprendizaje supervisado para la predicción de la ROP con el uso del lenguaje de programación de Python en la interfaz web de código abierto Jupyter Notebook del gestor de entorno Anaconda Navigator que permite la visualización y ejecución de código a través del navegador.

Debido a que se ha asumido un desgaste 1 de la broca con tendencia lineal y un desgaste 2 de la broca con tendencia radical, se considera crear un modelo con cada desgaste por separado junto a las otras 12 variables. Esto para comparar el desempeño de los modelos en la predicción de la ROP con cada tipo de desgaste.

Tabla 20. Dataset final

Variables	Tipo de variable
ROP	Numérica
Bit depth	Numérica
Hook load	Numérica
WOB	Numérica
Torque	Numérica
Bit RPM	Numérica
Pump	Numérica
Flow out	Numérica
Flow in	Numérica
ΔTemperature	Numérica
Rot time	Numérica
Overbalance	Numérica
Desgaste 1	Numérica
Desgaste 2	Numérica

Fuente: Autores.

3.4.1. Modelamiento. Con el uso de la biblioteca de aprendizaje automático Pycaret de código bajo y abierto en Python que permite pasar de preparar los datos a implementar un modelo, se usa la función compare_models y se generan las tablas 21 y 22 con los mejores modelos que se ajustan a los datos objeto de estudio.

Se puede observar que los seis modelos con mayor desempeño son: Extra Trees Regressor, Random Forest Regressor, Extreme Gradient Boosting, Catboost Regressor, K Neighbors Regressor y Light Gradient Boosting Machine.

Debido a que la plataforma Heroku que permite crear el microservicio tiene una capacidad máxima de 500 MB no se puede realizar el deployment con un modelo que exceda esta capacidad, por lo tanto, no habrá un enfoque en los modelos Extra trees Regressor, Random Forest Regressor y Catboost Regressor ya que su tiempo de aprendizaje es mayor y son más pesados. El enfoque estará en los modelos:

Extreme Gradient Boosting, K Neighbors Regressor y Light Gradient Boosting Machine.

Tabla 21. Comparación de las métricas de los modelos que mejor se ajustan al dataset con el desgaste 1

	Model	**MAE**	**RMSE**	**R2**	**TT(Sec)**
et	Extra Trees Regressor	0.7584	1.2975	0.7532	38.0560
rf	Random Forest Regressor	0.7947	1.3394	0.7371	92.8800
xgboost	Extreme Gradient Boosting	1.0018	1.5563	0.6456	10.1580
catboost	CatBoost Regressor	1.0361	1.5907	0.6301	23.5930
knn	K Neighbors Regressor	1.0249	1.6447	0.6044	1.0780
lightgbm	Light Gradient Boosting Machine	1.1060	1.6865	0.5844	1.3110
dt	Decision Tree Regressor	1.0347	1.8105	0.5201	1.4490
gbr	Gradient Boosting Regressor	1.3118	1.9534	0.4425	19.1920
lr	Linear Regression	1.6576	2.4848	0.0989	1.0090
ridge	Ridge Regression	1.6576	2.4848	0.0989	0.0860
br	Bayesian Ridge	1.6577	2.4848	0.0989	0.1960
lar	Least Angle Regression	1.6576	2.4848	0.0989	0.0780
en	Elastic Net	1.6663	2.4956	0.0911	0.1100

Fuente: Autores.

Tabla 22. Comparación de las métricas de los modelos que mejor se ajustan al dataset con el desgaste 2

	Model	MAE	RMSE	R2	TT(Sec)
et	Extra Trees Regressor	0.7605	1.2972	0.7533	42.1450
rf	Random Forest Regressor	0.7951	1.3387	0.7374	102.7080
xgboost	Extreme Gradient Boosting	1.0041	1.5583	0.6447	11.3270
catboost	CatBoost Regressor	1.0379	1.5979	0.6268	26.9110
knn	K Neighbors Regressor	1.0250	1.6450	0.6043	0.5610
lightgbm	Light Gradient Boosting Machine	1.1050	1.6874	0.5840	1.5950
dt	Decision Tree Regressor	1.0371	1.8178	0.5141	1.3990
gbr	Gradient Boosting Regressor	1.3046	1.9454	0.4469	22.8280
lr	Linear Regression	1.6581	2.4854	0.0985	0.7970
ridge	Ridge Regression	1.6582	2.4854	0.0985	0.0980
br	Bayesian Ridge	1.6577	2.4854	0.0985	0.2120
lar	Least Angle Regression	1.6631	2.4874	0.0971	0.0920
en	Elastic Net	1.6663	2.4956	0.0911	0.1090

Fuente: Autores.

3.4.2. División de datos de entrenamiento, prueba y validación. La división de estos datos fue realizada con Scikit learn, biblioteca de software de aprendizaje automático para el lenguaje de programación Python con la función train_test_split que permite fácilmente dividir un conjunto de datos de una matriz o DataFrame en dos aleatorios con un tamaño dado. No existe una forma directa de dividir el conjunto de datos en tres, pero es algo que se puede hacer anidando los valores de la siguiente manera:

- Se dividen los datos entre validación y entrenamiento que corresponde al 10% y 90% del conjunto original respectivamente.

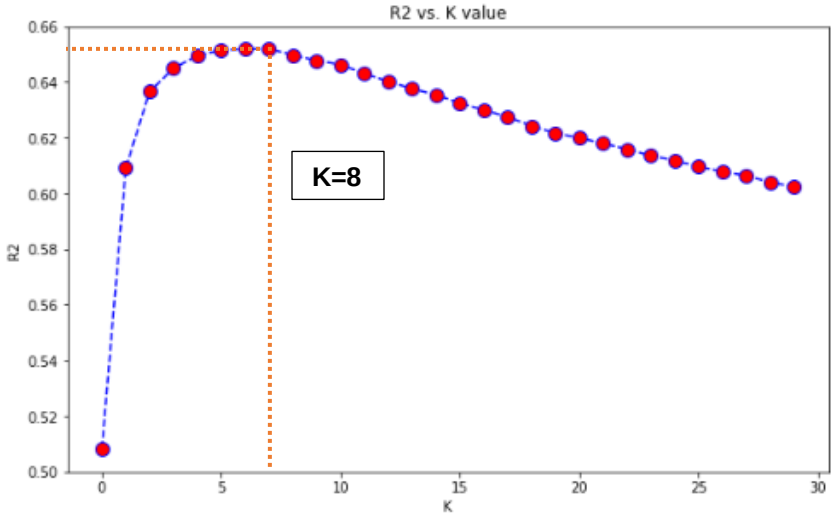
- Una vez hecho esto se divide el conjunto de entrenamiento en dos: 30% correspondiente a los datos de prueba y la otra parte conforma los datos de entrenamiento que corresponden al 60% del 90% del dataset original.

Generando tres conjuntos de datos en las proporciones 60/30/10. Teniendo en cuenta las divisiones realizadas, a continuación, se proceden a implementar los modelos de aprendizaje supervisado seleccionados.

3.4.3. K-Nearest Neighbor (KNN). Es importante seleccionar el valor correcto para K, por lo tanto, teniendo en cuenta el coeficiente de determinación (r2) y el error absoluto medio (MAE) se generan valores de estas métricas para diferentes K.

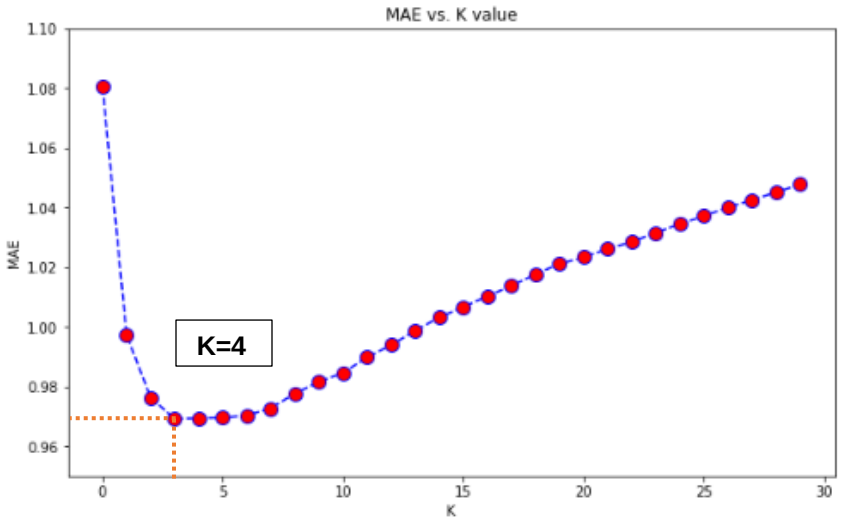
Las figuras 15 y 16 de R2 y MAE para los diferentes valores de K con el dataset del desgaste 1 indican que los valores óptimos de K con respecto a los mejores valores de R2 y MAE son: 8 y 4 respectivamente. Y las figuras 17 y 18 de R2 y MAE para los diferentes valores de K con el dataset del desgaste 2 muestran que los valores óptimos de K con respecto a los mejores valores de R2 y MAE son: 7 y 5 respectivamente.

Figura 15. R2 para diferentes valores de K en la base de datos con el desgaste 1



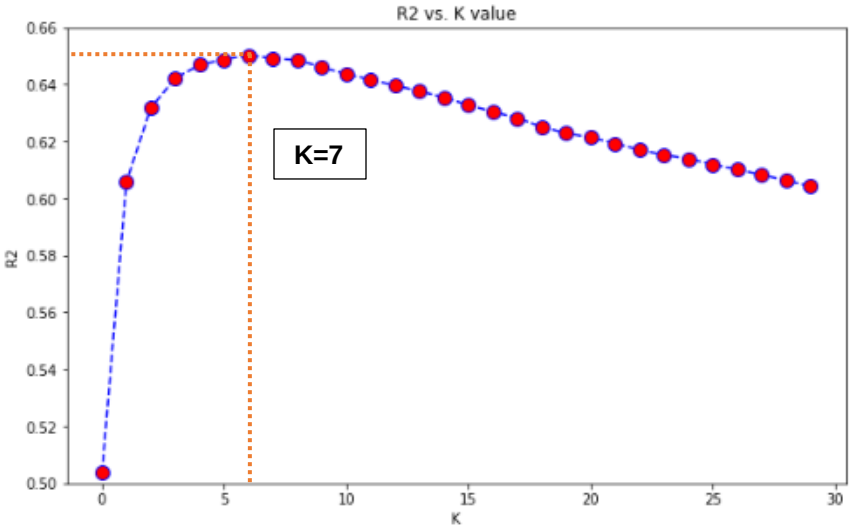
Fuente: Autores.

Figura 16. MAE para diferentes valores de K en la base de datos con el desgaste 1



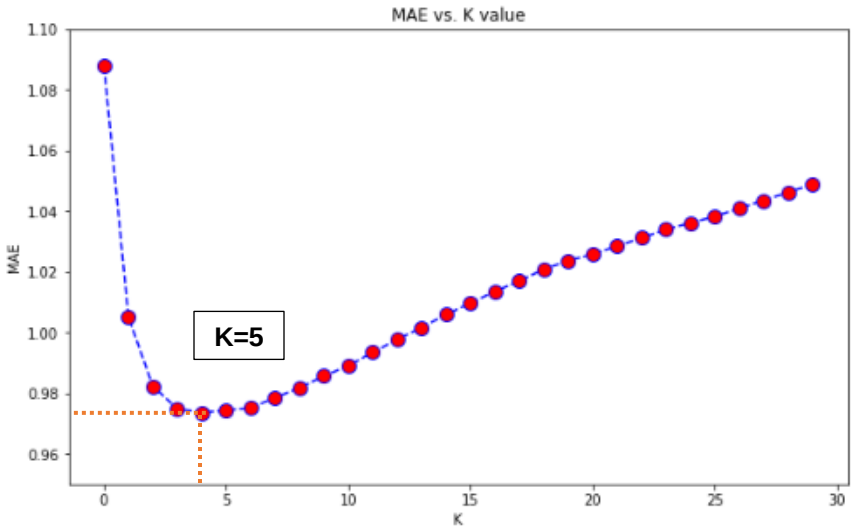
Fuente: Autores.

Figura 17. R2 para diferentes valores de K en la base de datos con el desgaste 2



Fuente: Autores.

Figura 18. MAE para diferentes valores de K en la base de datos con el desgaste 2



Fuente: Autores.

Ajuste de los hiperparámetros. La optimización de hiperparámetros es la acción de seleccionar el conjunto óptimo de hiperparámetros para un algoritmo de aprendizaje. Al determinar la combinación correcta, se maximiza el rendimiento del modelo, lo que significa que el algoritmo de aprendizaje toma mejores decisiones cuando se proporcionan instancias invisibles. Los valores seleccionados como hiperparámetros controlan el proceso de aprendizaje, por lo tanto, son diferentes de los parámetros normales ya que se seleccionan antes de entrenar un algoritmo de aprendizaje[45].

GridSearchCV es una clase disponible en scikit-learn que permite evaluar y seleccionar de forma sistemática los parámetros de un modelo. La búsqueda en cuadrícula consiste en buscar de forma exhaustiva a través de un subconjunto manual de valores específicos del espacio de hiperparámetros en un algoritmo de aprendizaje. Realizar una búsqueda en Grid significa que debe haber una métrica

[45] PYKES, Kurtis. Hyperparameter Optimization. Towards Data Science. 2010.

de rendimiento que guíe nuestro algoritmo. Este método considera exhaustivamente todas las combinaciones de parámetros.[46]

Se implementa el método de GridsearchCV para tunear el hiperparámetro del modelo KNN estimando el valor de K que mejor se ajusta a los datos con el fin de generar la optimización del modelo y maximizar su desempeño. Finalmente, esta técnica indica que K=6 es el valor óptimo para crear el modelo (tabla 23).

Tabla 23. Valores de K obtenidos como valores de K óptimos para el modelo KNN para cada dataset

Dataset Desgaste 1	Dataset Desgaste 2
K = 8	K = 7
K = 4	K = 5
K = 6	K = 6

Validación el modelo KNN. Al crear los modelos con cada valor del parámetro K en la tabla 23 se logra evidenciar que el K óptimo en los datos de validación es K=6 en los dos modelos (modelo con el dataset D1 y con el dataset D2) demostrando que el método GridSearchCV fue el más acertado. Se obtiene un valor de R2=0.6514 del modelo del dataset D1 y R2=0.6487 del modelo del dataset D2 con los datos de prueba y con los datos de validación se obtiene un desempeño con un R2=0.6035 del modelo con el dataset D1 y un R2=0.6023 del modelo con el dataset D2 como se puede observar en las tablas 24 y 25.

Tabla 24. Resultados de R2 y MAE del modelo de KNN de cada dataset en los datos de prueba

	Dataset D1	Dataset D2
R2	0.65144225	0.64874022
MAE	0.96950372	0.97425883

[46] RANJAN, G S K et. al. K-Nearest Neighbors and Grid Search CV Based Real Time Fault Monitoring System for Industries. 2019.

Tabla 25. Resultados de R2 y MAE del modelo de KNN de cada dataset en los datos de validación

	Dataset D1	Dataset D2
R2	0.60349765	0.60225071
MAE	0.97475923	0.97793829

3.4.4. Light Gradient Boosting Machine (LightGBM)

Parámetros del modelo LightGBM. En la tabla 26 se denotan los hiperparámetros del modelo LightGBM con su respectivo significado y uso.

Tabla 26. Hiperparámetros del modelo LightGBM

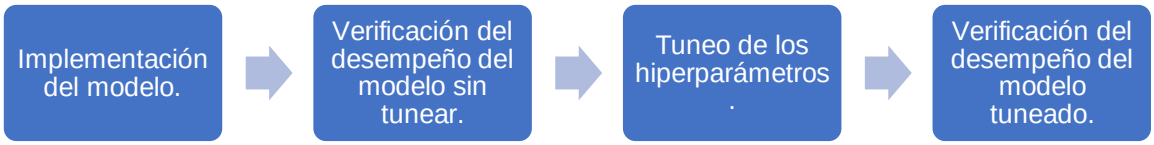
Parámetro	Significado	Uso
max_depth	Profundidad máxima del árbol.	Se usa para manejar el sobreajuste del modelo. Si el modelo está sobreajustado, se debe reducir max_depth.
feature_fracti on	Subconjunto de características.	LightGBM seleccionará aleatoriamente un subconjunto de características en cada nodo del árbol si feature_fraction_bynode es más pequeño que 1.0. Por ejemplo, si lo configura en 0.8, LightGBM seleccionará el 80% de las características en cada nodo del árbol.
bagging_fract ion	Fracción de datos que se usará para cada iteración.	Se usa para acelerar el entrenamiento y evitar el sobreajuste.

bagging_freq	Frecuencia de bagging.	0 significa deshabilitar el bagging; k significa realizar bagging en cada k iteración. Para habilitar el ensacado, bagging_fraction debe establecerse en un valor menor que 1.0
lambda	El valor típico varía de 0 a 1.	Especifica la regularización.
min_child_samples	Número mínimo de datos en una hoja.	Puede usarse para tratar el sobreajuste.
objective	Especificación de la aplicación del modelo, ya sea un problema de regresión o un problema de clasificación.	Se considera por defecto el modelo como un modelo de regresión.
learning_rate	Tasa de aprendizaje.	Determina el impacto de cada árbol en el resultado final. Funciona comenzando con una estimación inicial que se actualiza utilizando la salida de cada árbol. También afecta a los pesos de normalización de los árboles caídos
num_leaves	Número máximo de hojas en un árbol.	Por defecto: 31
metric	Métrica(s) a evaluar en el(los) conjunto(s) de evaluación.	Especifica la pérdida para la construcción del modelo.

Fuente: DOWNEY, A.B. How to Think Like a Computer Scientist. Learning with Python. Green Tea Press. 2008.

Implementación del modelo. De acuerdo con el funcionamiento de LightGBM se procede a implementar el modelo con los parámetros por defecto para probar su predicción con los datos de prueba y con los datos de validación. Teniendo en cuenta lo anterior, se llevan a cabo las etapas descritas en la figura 19.

Figura 19. Etapas de la implementación del modelo de aprendizaje automático



Antes de implementar el modelo es importante convertir los datos de entrenamiento, prueba y validación en una matriz LightGBM optimizada para mayor eficiencia. Después, con el uso de la biblioteca de Scikit learn y con la función LGBMRegressor() se entrena el modelo base. Seguido de esto se realiza el tuneo de los hiperparámetros del modelo para optimizarlo y conseguir un mayor desempeño en la predicción.

Ajuste de los hiperparámetros. Con el uso del marco de software de optimización de hiperparámetros automático Optuna, especialmente diseñado para el aprendizaje automático, se realiza la búsqueda de los hiperparámetros óptimos para el modelo. Optuna es un framework diseñado para la automatización y la aceleración de los estudios de optimización en donde se usan los términos *study* y *trial,* refiriéndose a *study* como el estudio donde se realiza la optimización basada en una función objetivo y un *trial* se refiere a una ejecución. En este caso se define una función que depende de n_trial en donde también se indican los parámetros a optimizar y retorna el valor del coeficiente de determinación r2. Seguido de esto, se llama el estudio de optuna con dirección a maximizar el r2 para optimizar la función objetivo con 1000 pruebas (n_trial=1000).

Una vez realizada la optimización, con *study.best_trial* se generan los parámetros que permiten obtener el mejor r2 en la optimización (tablas 27 y 28).

Tabla 27. Ajuste de hiperparámetros del modelo LightGBM con el dataset D1

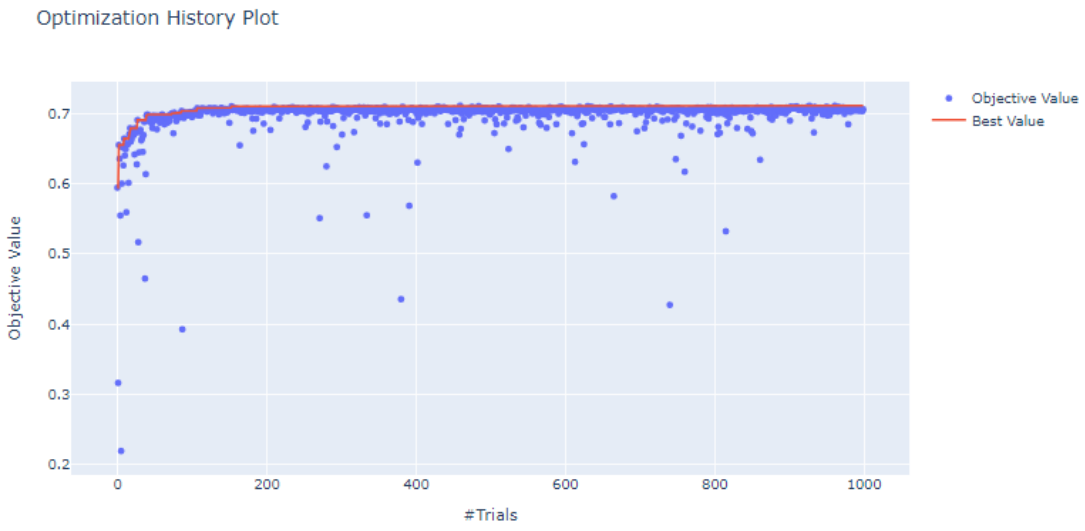
Paramétros	**Valor óptimo**
max_depth	20
feature_fraction	0.935148944818218
bagging_fraction	0.9993524960699982
bagging_freq	6
lambda_l1	1.699648164766365
lambda_l2	0.9373736961109946
min_child_samples	5
learning rate	0.2650351193128691
num_leaves	248

Tabla 28. Ajuste de hiperparámetros del modelo LightGBM con el dataset D2

Paramétros	**Valor óptimo**
max_depth	19
feature_fraction	0.9661447083127549
bagging_fraction	0.9828922823572798
bagging_freq	6
lambda_l1	6.459606852751049e-07
lambda_l2	2.0487143576431053
min_child_samples	9
learning rate	0.2387770401022269
num_leaves	245

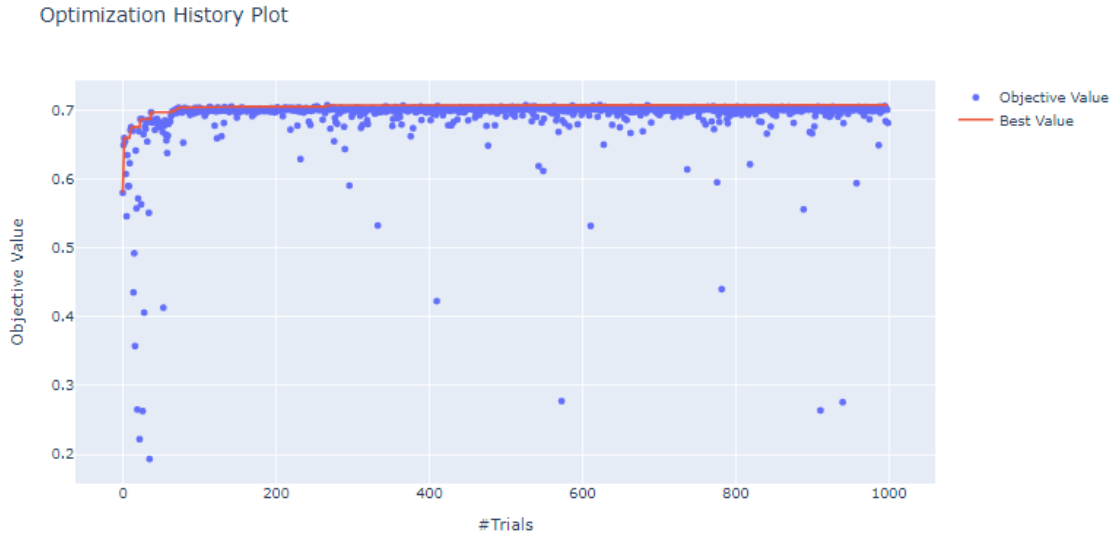
La figura del historial de optimización de los hiperparámetros del modelo LightGBM presenta los valores objetivo que se obtuvieron representados con puntos azules y con la línea roja se indica la tendencia de los mejores estimadores (Figura 20 con el dataset D1 y Figura 21 con el dataset D2). Se evidencia el comportamiento de cada parámetro del modelo en el proceso de optimización demostrando la tendencia ascendente debido a la maximización de la métrica R2.

Figura 20. Gráfico del historial de optimización del modelo LightGBM con el dataset D1



Fuente: Autores.

Figura 21. Gráfico del historial de optimización del modelo LightGBM con el dataset D2



Fuente: Autores.

Finalmente, se realiza la validación del modelo de aprendizaje automático LightGBM comparando las métricas del modelo sin tuneo y con el tuneo de los hiperparámetros.

Tabla 29. Evaluación del coeficiente de determinación R2 con los datos de test en los modelos sin tunear

Modelo base		
Métrica	**Dataset D1**	**Dataset D2**
R2	0.60041443	0.60070398

Tabla 30. Evaluación del coeficiente de determinación R2 con los datos de test en los modelos tuneados

Modelo tuneado		
Métrica	**Dataset D1**	**Dataset D2**
R2	0.71139335	0.70812596

Tabla 31. Evaluación de las métricas R2 y MAE con los datos de validación en los modelos tuneados

Datos de validación		
Métrica	**Dataset D1**	**Dataset D2**
R2	0.66520858	0.66858746
MAE	0.89242166	0.89688486

Las tablas 29, 30 y 31 representan los valores de las métricas obtenidas por los modelos LightGBM donde se observa claramente que el tuneo de los hiperparámetros permite que aumente más de un 10% el coeficiente de determinación R2 en los datos de prueba y aumente más de 6% en los datos de validación. Por lo tanto, se logra crear el modelo de predicción Light Gradient Boosting Machine del dataset D1 con un coeficiente de determinación R2 del 66.52% y un error absoluto medio MAE de 0.8924 y el modelo creado con el dataset D2 logra llegar a un 66.86% de coeficiente de determinación y con un error absoluto medio de 0.8969.

3.4.5. Extreme Gradient Boosting (XGBoost)

Parámetros del modelo XGBoost. En la tabla 32 se denotan los hiperparámetros del modelo XGBoost con su respectivo significado y uso.

Tabla 32. Parámetros del modelo XGBoost

Parámetro	Significado	Uso
max_depth	"Profundidad" o número de nodos de bifurcación de los árboles de	Aunque una mayor profundidad puede devolver mejores resultados, también puede

	decisión usados en el entrenamiento.	resultar en overfitting o sobreajuste.
reg_alpha	Plazo de regularización L1 sobre ponderaciones.	Incrementar este valor hará que el modelo sea más conservador.
reg_lambda	Plazo de regularización L2 sobre ponderaciones.	Incrementar este valor hará que el modelo sea más conservador.
min_child_weight	Suma mínima de peso de instancia (arpillera) necesaria en un child. En la tarea de regresión lineal, esto simplemente corresponde al número mínimo de instancias necesarias para estar en cada nodo.	Si el paso de la partición del árbol da como resultado un nodo hoja con la suma del peso de la instancia menor que min_child_weight, entonces el proceso de construcción dejará de particionar. Cuanto más grande min_child_weight sea, más conservador será el algoritmo.
gamma	Rango: [0, ∞]	Se requiere una reducción mínima de pérdidas para realizar una nueva partición en un nodo hoja del árbol. Cuanto más grande gamma sea, más conservador será el algoritmo.
colsample_bytree	Es la proporción de submuestra de columnas al construir cada árbol.	El submuestreo ocurre una vez por cada árbol construido.

objective	El tipo de tarea de clasificación que se realiza.	Especificación de la aplicación del modelo, ya sea un problema de regresión o un problema de clasificación.
learning_rate	Reducción del tamaño del paso utilizada en la actualización para evitar el sobreajuste.	Después de cada paso de impulso, podemos obtener directamente los pesos de las nuevas funciones y reducirlos para hacer que el proceso de impulso sea más conservador.
colsample_bylevel	Es la proporción de submuestras de columnas para cada nivel.	El submuestreo ocurre una vez por cada nuevo nivel de profundidad alcanzado en un árbol. Las columnas se submuestrean del conjunto de columnas elegido para el árbol actual.
colsample_bynode	Es la proporción de submuestra de columnas para cada nodo (división).	El submuestreo ocurre una vez cada vez que se evalúa una nueva división. Las columnas se submuestrean del conjunto de columnas elegido para el nivel actual.
subsample	Proporción de submuestras de las instancias de formación.	Establecerlo en 0.5 significa que XGBoost muestreará al azar la mitad de los datos de entrenamiento antes de cultivar árboles y esto evitará el sobreajuste. El submuestreo

		ocurrirá una vez en cada iteración de impulso.
scale_pos_weight	Controla el balance de ponderaciones positivas y negativas.	Útil para clases no balanceadas.

Fuente: DOWNEY, A.B. How to Think Like a Computer Scientist. Learning with Python. Green Tea Press. 2008.

Implementación del modelo. Teniendo en cuenta el proceso del funcionamiento de XGBoost se procede a cumplir con las etapas de la implementación del modelo de aprendizaje automático de la figura 19. Adicionalmente es importante convertir los datos de entrenamiento, prueba y validación en una matriz XGBoost optimizada para mayor eficiencia. Con el uso de la biblioteca de Scikit learn y con la función XGBRegressor se entrena el modelo base. Seguido de esto se realiza el tuneo de los hiperparámetros del modelo XGBoost para optimizarlo y conseguir un mayor desempeño del modelo.

Ajuste de los hiperparámetros. En este caso se define la función objetivo con los parámetros a optimizar y retorna el error cuadrático medio (RMSE) que mide la cantidad de error que hay entre dos conjuntos de datos, en otras palabras, compara un valor predicho y un valor observado o conocido. Después, se llama el estudio de optuna con dirección a minimizar el RMSE para optimizar la función objetivo con 1000 pruebas (n_trial=1000). Una vez realizada la optimización, con *study.best_trial* se generan los parámetros que permiten obtener el menor RMSE en la optimización (tablas 33 y 34).

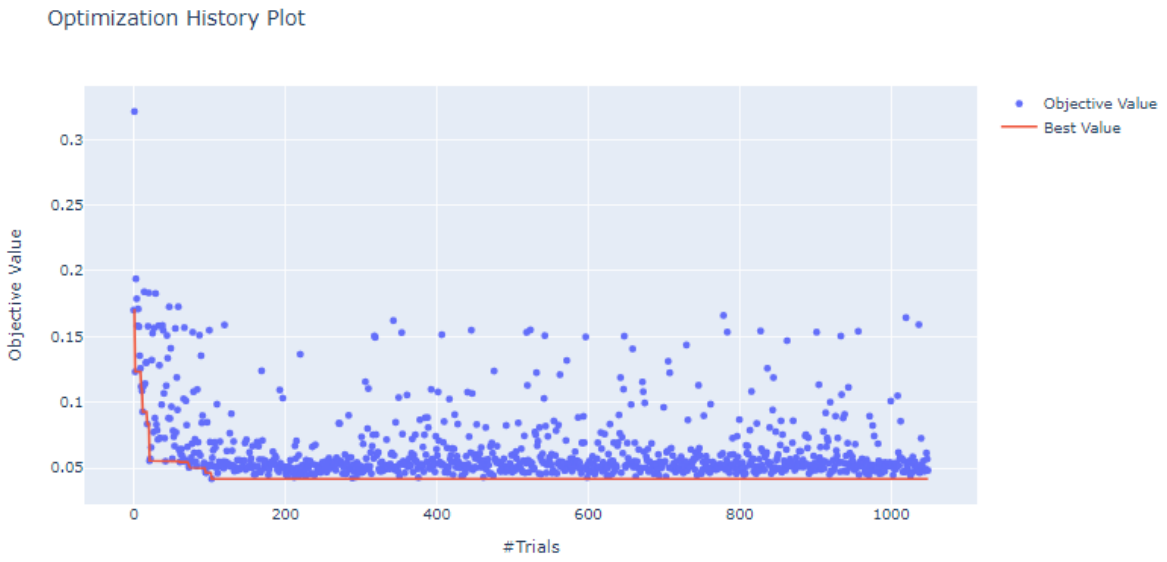
Tabla 33. Ajuste de hiperparámetros del modelo XGBoost con el dataset D1

Paramétros	Valor óptimo
max_depth	20
reg_alpha	0.0024988692989273394
reg_lambda	0.1258661835935828
min_child_weight	0
gamma	0.0002352732263283589
colsample_bytree	0.8651439096981715
colsample_bylevel	0.851952289057502
colsample_bynode	0.39745354596026866
subsample	0.9847558902045351
learning_rate	0.02109680955533315
scale_pos_weight	0.18918097012704807

Tabla 34. Ajuste de hiperparámetros del modelo XGBoost con el dataset D2

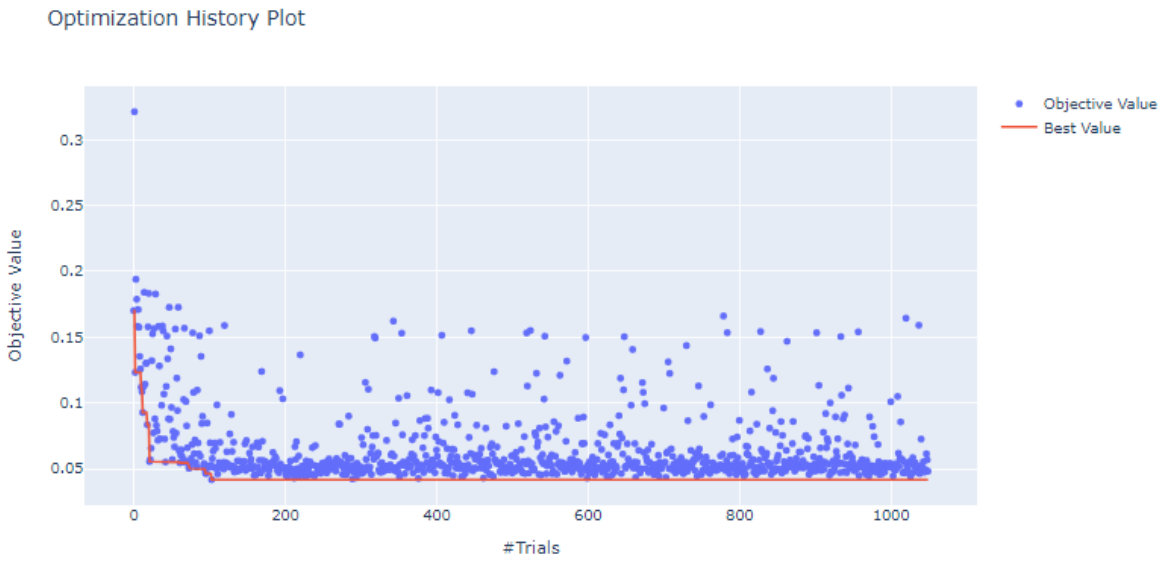
Paramétros	Valor óptimo
max_depth	18
reg_alpha	0.24854949270756252
reg_lambda	0.0023945094638918125
min_child_weight	0
gamma	0.061653033669902546
colsample_bytree	0.7998843525323054
colsample_bylevel	0.8006760035147253
colsample_bynode	0.4961529274602456
subsample	0.9132085055363002
learning_rate	0.011168297294091281
scale_pos_weight	0.659201665882062

Figura 22. Gráfico del historial de optimización del modelo XGBoost con el dataset D1



Fuente: Autores.

Figura 23. Gráfico del historial de optimización del modelo XGBoost con el dataset D2



Fuente: Autores.

El gráfico del historial de optimización de los hiperparámetros del modelo XGBoost presenta los valores objetivo RMSE representados con puntos azules y con la línea roja se indica la tendencia de los mejores estimadores (Figuras 22 y 23). Se representa el comportamiento de cada parámetro del modelo en el proceso de optimización del modelo demostrando la tendencia descendente debido a la minimización del RMSE.

Finalmente, se procede a realizar la validación del modelo de aprendizaje automático XGBoost comparando las métricas R2 y RMSE del modelo sin tuneo y con el tuneo de los hiperparámetros.

Tabla 35. Evaluación del coeficiente de determinación R2 y el RMSE con los datos de test en los modelos sin tunear

Modelo base		
Métrica	**Dataset D1**	**Dataset D2**
R2	0.66290752	0.66066323
RMSE	1.389005	1.459601

Tabla 36. Evaluación del coeficiente de determinación R2 y el RMSE con los datos de test en los modelos tuneados

Modelo tuneado		
Métrica	**Dataset D1**	**Dataset D2**
R2	0.77441436	0.77095528
rmse	1.231529	1.240895

Tabla 37. Evaluación de las métricas R2 y MAE con los datos de validación en los modelos tuneados

Datos de validación		
Métrica	**Dataset D1**	**Dataset D2**
R2	0.74426703	0.73454225
MAE	0.72704637	0.75031145

Las tablas 35, 36 y 37 representan los valores de las métricas obtenidas por los modelos XGBoost donde se puede observar que el tuneo de los hiperparámetros permite que aumente más de un 11% el coeficiente de determinación R2 en los datos de prueba y aumente más de 7% en los datos de validación. Por lo tanto, se logra crear el modelo de predicción Extreme Gradient Boosting del dataset D1 con un coeficiente de determinación R2 del 74.43% y un error absoluto medio MAE de 0.7270 y el modelo creado con el dataset D2 logra llegar a un 73.45% de coeficiente de determinación r2 y con un error absoluto medio de 0.7503.

3.4.6. Evaluación. De acuerdo con la metodología CRISP-DM, se continúa con la fase de evaluación de los modelos. En esta etapa del proyecto ya se han construido 3 modelos (KNN, LightGBM y XGBoost) que presentan alta calidad desde la perspectiva de análisis empleada. Antes de proceder a desplegar y poner en uso el microservicio, es de vital importancia evaluar los modelos y revisar todos los pasos ejecutados en su construcción para verificar apropiadamente que este cumple con los objetivos del negocio. Al final de esta fase, se debe decidir si los resultados del proceso de minería de datos están listos para su uso.

La métrica usada para evaluar la calidad de los modelos es el coeficiente de determinación R2, el cual nos indica que el modelo XGBoost tiene un mayor desempeño en la predicción de la ROP en la formación objeto de estudio, seguido del modelo LightGBM y con el menor nivel de predictibilidad entre los modelos implementados está el modelo KNN, como se puede observar en las tablas 38 y 39.

Tabla 38. Coeficientes de determinación de los modelos creados con el Dataset D1

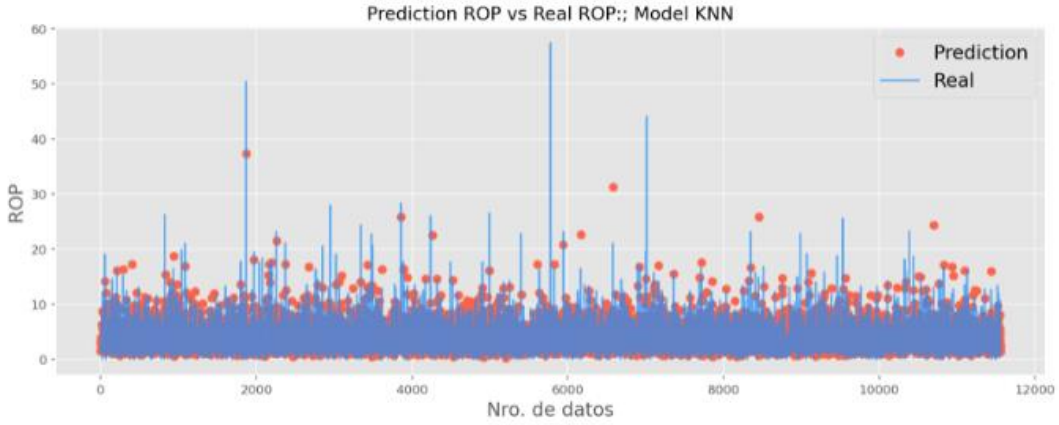
	KNN	LightGBM	XGBoost
R2	0.6035	0.6652	0.7443

Tabla 39. Coeficientes de determinación de los modelos creados con el Dataset D2

	KNN	LightGBM	XGBoost
R2	0.6022	0.6686	0.7345

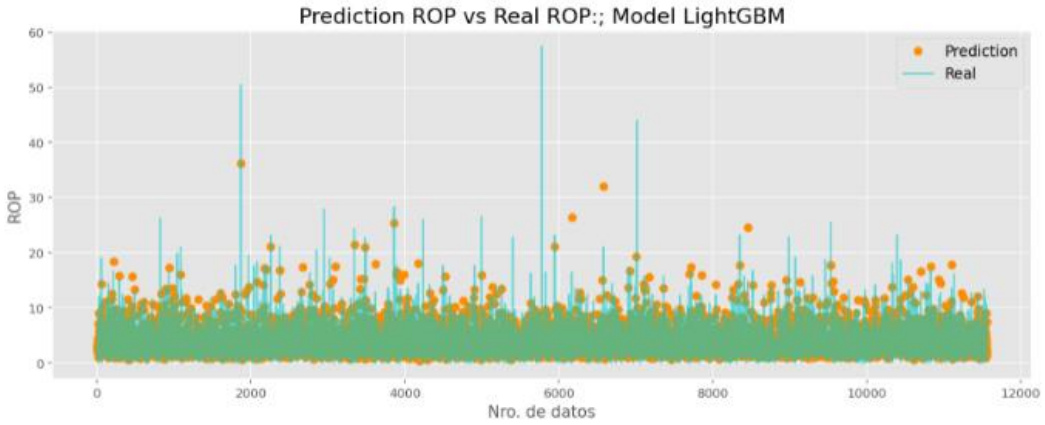
Con los datos de validación se evalúa la capacidad de generalización del modelo predictivo, obteniendo los resultados indicados en las tablas anteriores. Además, se puede observar en las siguientes gráficas la comparación de los valores reales de ROP y los valores predichos:

Figura 24. Validación del modelo de predicción KNN



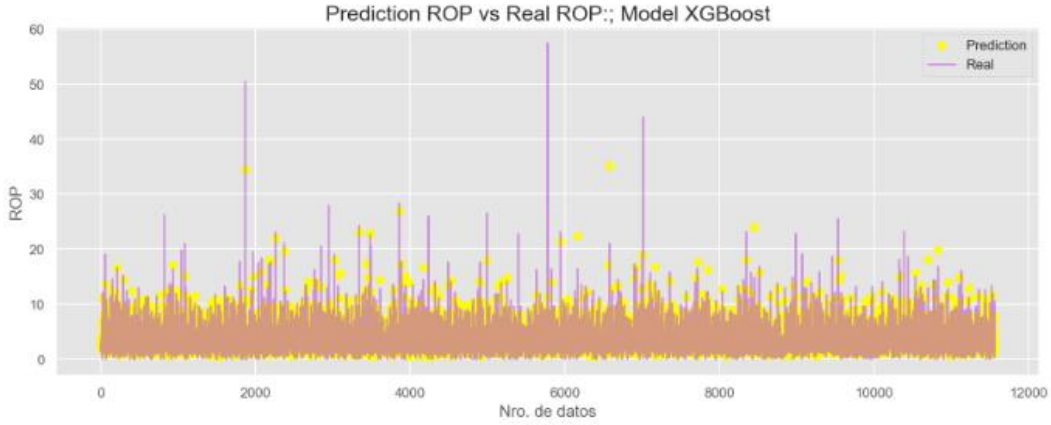
Fuente: Autores.

Figura 25. Validación del modelo de predicción LightGBM



Fuente: Autores.

Figura 26. Validación del modelo de predicción XGBoost



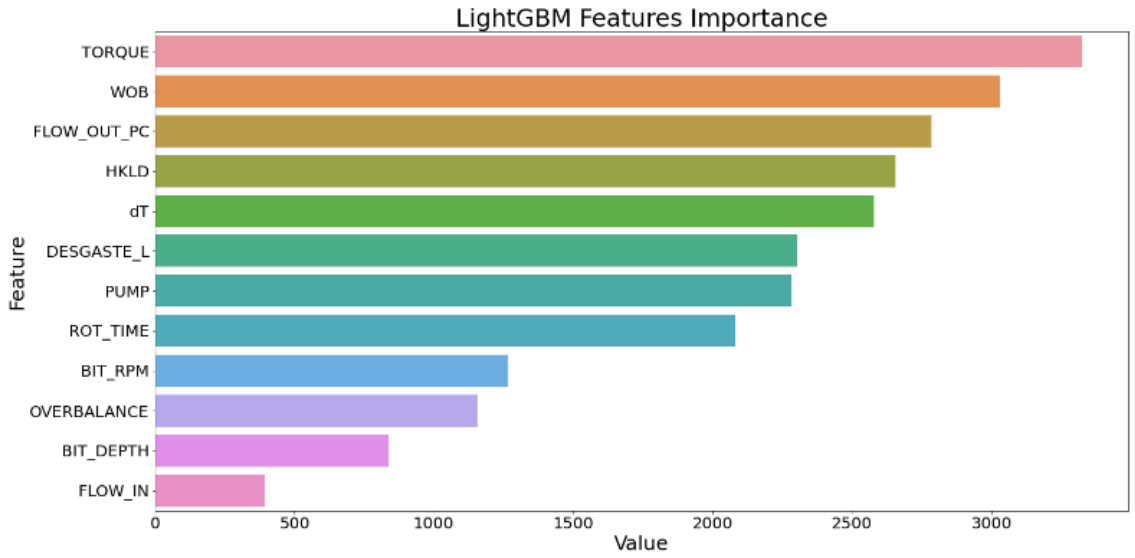
Fuente: Autores.

Las figuras 24, 25 y 26 muestran el comportamiento de los datos reales (representados con una línea) y las predicciones realizadas por los modelos (representadas con puntos) evidenciando el desempeño obtenido principalmente para valores de ROP bajas, en su mayoría menores a 20 ft/hr; en valores más altos de ROP se obtiene un menor desempeño en la predicción. Esto debido a que la

mayoría de los valores de ROP se concentran en el rango de 0 a 20 ft/hr y a la complejidad del comportamiento de esta variable, en especial en formaciones específicas caracterizadas por ser duras y abrasivas como la formación objeto de estudio.

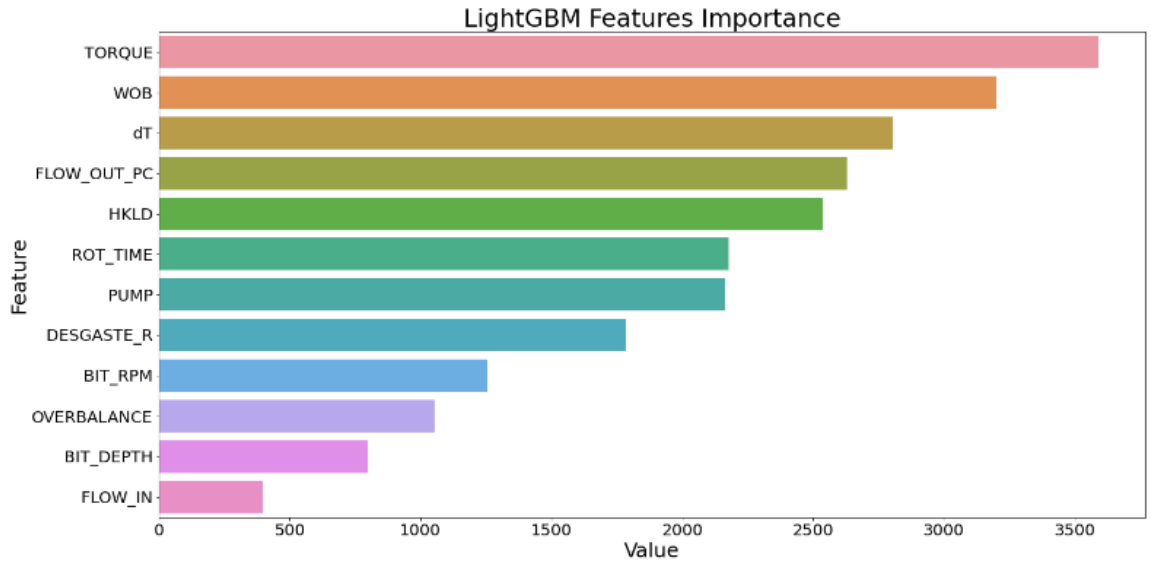
Debido a que la ROP no tiene una correlación lineal con las demás variables, se usa la técnica *feature_importances_* de Scikit learn, que asigna una puntuación a las características de entrada en función de su utilidad para predecir una variable objetivo, aprovechando la capacidad que tienen los modelos asociados con árboles de decisión (en este caso, LightGBM y XGBoost) de capturar la relación no lineal entre las variables. El atributo ajustado *feature_importances_* se calcula como la media y la desviación estándar de la acumulación de la disminución de impurezas dentro de cada árbol.

Figura 27. Importancia de las características del modelo LightGBM con el dataset D1



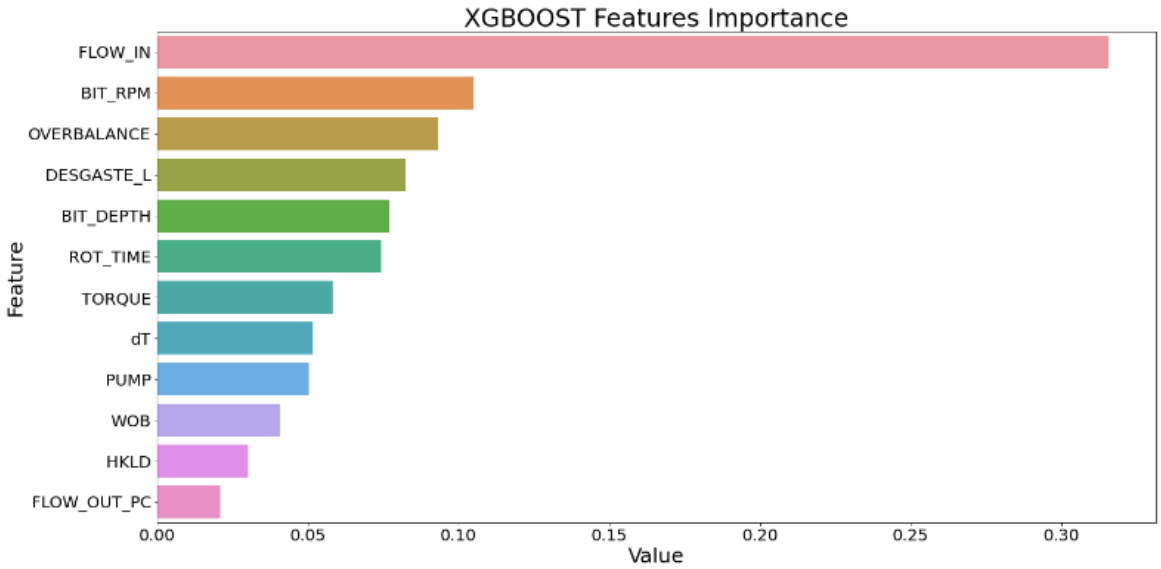
Fuente: Autores.

Figura 28. Importancia de las características del modelo LightGBM con el dataset D2



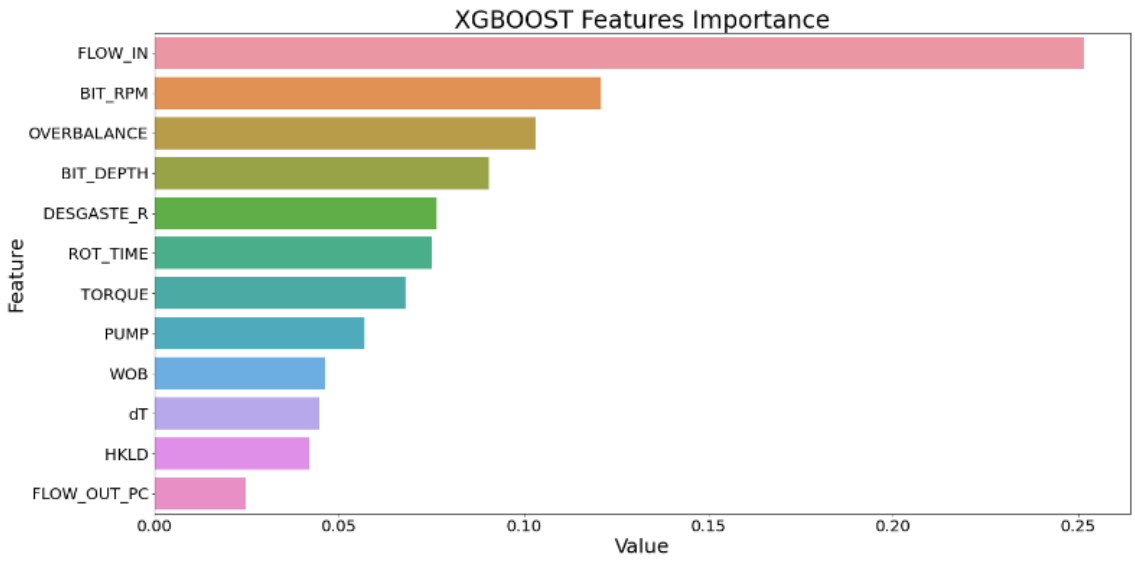
Fuente: Autores.

Figura 29. Importancia de las características del modelo XGBoost con el dataset D1



Fuente: Autores.

Figura 30. Importancia de las características del modelo XGBoost con el dataset D2



Fuente: Autores.

De acuerdo con las gráficas 27, 28, 29 y 30 se puede observar que el caudal es la variable con menor relevancia o aporte al modelo LightGBM, pero ocurre todo lo contrario con XGBoost ya que en este modelo demuestra ser la variable con mayor impacto en la predicción de la ROP. El overbalance también presenta poca incidencia en el LightGBM y en el modelo XGBoost ocupa el tercer lugar de importancia. El desgaste con tendencia lineal tiene más importancia que el desgaste con tendencia radical en los dos modelos de predicción.

Después de analizar el desempeño de los modelos verificando su calidad y el impacto de cada variable en la predicción de la ROP, se permite proceder a la fase del uso y despliegue del modelo.

3.5. DEPLOYMENT

Como etapa final del proyecto se procede a la realización de un microservicio que sea de utilidad para aquellos que requieran de su uso, que cuente con la capacidad de predecir la ROP usando los datos de entrada estipulados en la base de datos final con la que se realizó el modelo.

Para la realización del "deployment" o puesta en producción del proyecto se debieron tener en cuenta diferentes aspectos, como ¿Qué modelo de machine learning se debe usar?, ¿qué herramienta de programación web permite poner en marcha un proyecto de análisis de datos? Y, además, ¿en qué servidores se podría albergar luego de ya ser desarrollado? En este caso se implementa el modelo realizado Light Gradient Boosting Machine debido a que es el modelo que cumple tanto con los requerimientos de capacidad impuesto por el servidor a usar frente al modelo XGBoost y con la mejor predictibilidad respecto al modelo KNN.

Con respecto a la herramienta de programación web, se utiliza la librería de Python llamada "Streamlit", en esta se realiza toda la programación y se escribe el código para lanzar el microservicio de manera local, posteriormente para cumplir con el objetivo de poner en uso para cualquier persona que pueda acceder desde otro dispositivo ya sea móvil o cualquiera que cuente con internet, se utiliza la plataforma gratuita "Heroku" la cual permite el desarrollo de la app mediante una conexión con el servicio "GIT HUB" como se apreciará a continuación.

3.5.1. Streamlit. En primer lugar, se desarrolla la interfaz de la aplicación web mediante el servicio de la librería "Streamlit" de Python, se hizo la instalación de la librería usando la consola de comandos de Windows. Teniendo ya las librerías correspondientes instaladas se procede a desarrollar el código para el desarrollo de la aplicación, se crea una carpeta nueva donde se almacena el primer archivo llamado ***app.py.*** El primer paso fue importar todas las librerías necesarias para que

el programa funcionara de manera adecuada, teniendo en cuenta las librerías utilizadas en las secciones anteriores.

Se añaden las librerías ***pickle*** y ***joblib***, que serán útiles a la hora de leer archivos con extensión ***.pkl***, extensión que se usó para finalizar el modelo predictivo y el siguiente paso consta de cargar los modelos con desgaste lineal y desgaste radial para así poder usarlos dentro de la interfaz a la hora de predecir.

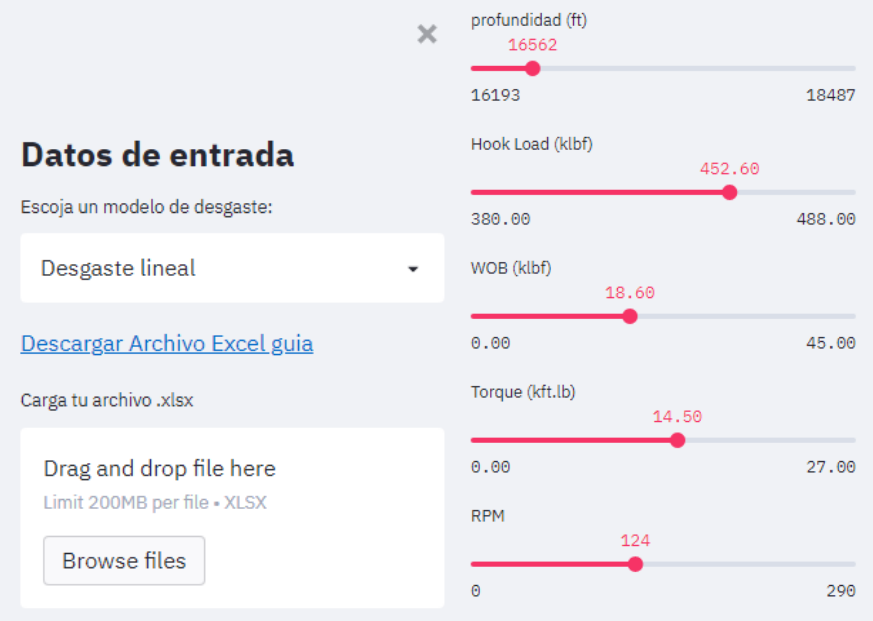
A través del código, se usan las funciones ***st.write , st.subheader*** y ***st.title*** las cuales son útiles para agregar texto dentro de la aplicación web, ambos tienen la misma funcionalidad, sin embargo, ***subheader,*** como su nombre lo indica, pone subtítulos con mayor tamaño y en negrilla. La función ***st.write*** puede leer prácticamente cualquier otra función de las librerías de Python como gráficas, DataFrames, entre otros.

Otra característica que se usó de la librería de ***streamlit*** es ***st.sidebar*** la cual permite poner información en una barra lateral desplegable. Para este proyecto, esta barra se utilizó como la sección en la cual el usuario inserta las entradas para el modelo, es decir, mediante barras deslizables llamadas sliders, se puede realizar cambios variables en los parámetros de entrada. Además, en la barra lateral, se decidió agregar no solo datos de entrada analógicos, sino también mediante el cargue de un archivo ***.xlsx*** que contenga ya los parámetros especificados. En cuanto el usuario tenga la plantilla, que podrá descargar mediante un enlace seguro, el paso a seguir es cargar ese archivo y se obtendrá la predicción de aquellos parámetros introducidos (figura 31).

En el código pertinente a la elaboración de la barra lateral, se decidieron poner los parámetros que se usaron en la predicción del modelo ya que son necesarios todos para poder obtener una predicción correcta, además, se hizo un diccionario, que permitiera asociar diferentes palabras clave con su respectivo valor. También se hace uso de la función ***st.sidebar.slider*** para asociar los valores en lo que se va a

acotar el slider y el valor predeterminado cuando se inicial la app o el valor por defecto.

Figura 31. Barra lateral aplicación web y método de introducción de datos



Los parámetros de entrada están acotados por los valores predeterminados donde mejor funciona la predicción. Antes de realizar cualquier predicción, se debe elegir el tipo de desgaste predilecto, la aplicación elegirá uno de los dos por defecto. Luego de haber elegido todos los parámetros con los que se desea realizar el estudio y el modelo de desgaste que se usará, se puede lanzar el predictor, este arrojará un valor específico, el cual tendrá un valor aproximado de predicción de ***± 0.9*** correspondiente al error absoluto medio del modelo (figura 32). En los casos en los que se adicionen más de una predicción, es decir, el archivo ***.xlsx*** contenga más de 1 fila, el predictor permitirá ver una gráfica expuesta con la librería plotly (figura 33), la cual permitirá ver las diferentes tasas de penetración, a medida que varía la profundidad de perforación.

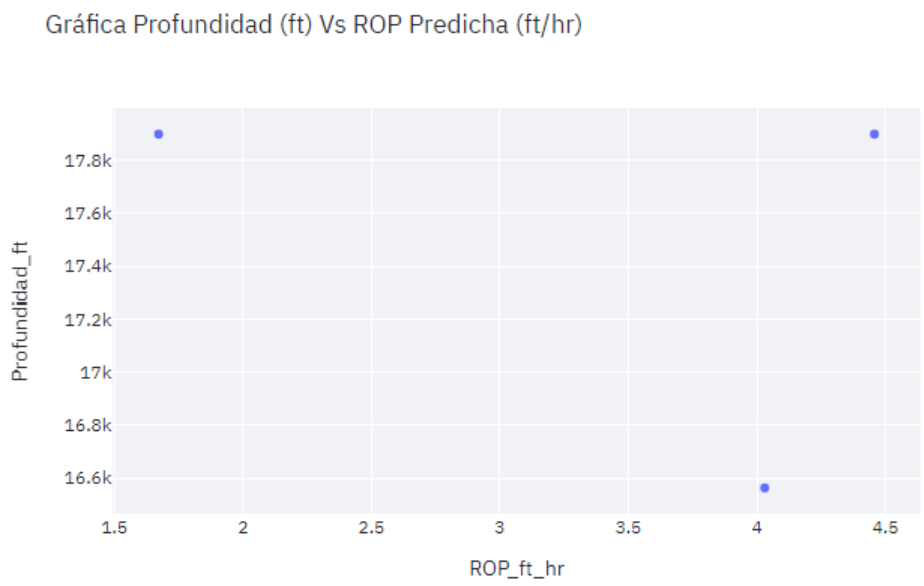
El lanzamiento del predictor se decidió exponer mediante un botón con la función ***st.botton***, este se enlaza mediante el modelo respectivo, que con la función ***.predict*** se obtendrá ya sea un dataframe o un valor concreto, dependiendo del método de uso que se utilice, ya sea con las barras o con el cargue del archivo.

Finalmente, luego de tener todo el código correctamente escrito, se procede a lanzar la aplicación en el servidor local de la computadora que se esté usando, mediante el comando **streamlit run app.py**.

A partir de este punto, se decidió que la aplicación de forma local funcionaba de manera adecuada, las características expuestas anteriormente trabajaban de forma correspondiente y cumplían sus funciones, por lo tanto, se procedió a desarrollar este en un servidor externo.

Figura 32. Resultado de ejecución de la aplicación web con parámetros deslizables

Figura 33. Gráfica Profundidad Vs. Predicción generada por la aplicación



3.5.2. GitHub. GitHub es una plataforma que permite el alojamiento de software y proporciona herramientas útiles para el trabajo en equipo[47]. Esta plataforma es de uso gratuito, los repositorios de código compartido pueden ser públicos o privados, para este proyecto se hará uso de las herramientas gratuitas. Los servicios de esta plataforma serán útiles para este proyecto, ya que es necesario crear un repositorio para conectar con el servidor externo, en este caso Heroku, pero para ello se deben tener en cuenta diferentes pautas a seguir.

En primera instancia se debe crear una carpeta que contenga:

- Código de streamlit, en este caso se llamará ***app.py***, es necesario que sea guardado con el nombre de la extensión, en este caso ***.py (Python)***
- Debe contener los modelos usados dentro del código, con su respectiva extensión, en este proyecto se usaron 2 modelos con dos funciones de desgaste diferentes.
- En la carpeta principal, debe estar incluido un archivo cuyo nombre debe ser procfile, este contiene un archivo de texto con la orden de ejecutar streamlit

[47] CASTILLO, Luciano. Conociendo GITHUB Documentation. 2017.

como se vió en la sección anterior y el nombre del archivo que contiene el código.

```
web: sh setup.sh && streamlit run app.py
```

- Como parte fundamental, un archivo de texto cuyo nombre debe ser requirments.txt, este se debe adicionar a la carpeta con el fin de agregar las librerías necesarias para que funcione la aplicación, seguido por un igual y su respectiva versión, este es un paso importante ya que al momento de realizar el despliegue en Heroku, la plataforma instala aquellas librerías con las versiones en el servidor externo para permitir su uso.

```
joblib==1.0.1
Flask==1.1.2
numpy==1.19.5
pandas==1.2.4
streamlit==0.84.1
scikit-learn==0.23.2
lightgbm==3.2.1
plotly==5.1.0
xlrd==2.0.1
openpyxl==3.0.7
```

- En última instancia se agregó un archivo de nombre setup, este archivo de texto debe contener el siguiente código:

```
mkdir -p ~/.streamlit/

echo "\
[server]\n\
port = $PORT\n\
enableCORS = false\n\
headless = true\n\
\n\
" > ~/.streamlit/config.toml
```

- Un último archivo que no es indispensable, pero sin embargo se recomienda adicionar es la versión de Python en la cual funciona y fue diseñado el código de la aplicación, en este caso se usó la versión 3.8.8, este archivo de texto debe llamarse runtime y la versión debe ir escrita de la siguiente forma:

```
python-3.8.8
```

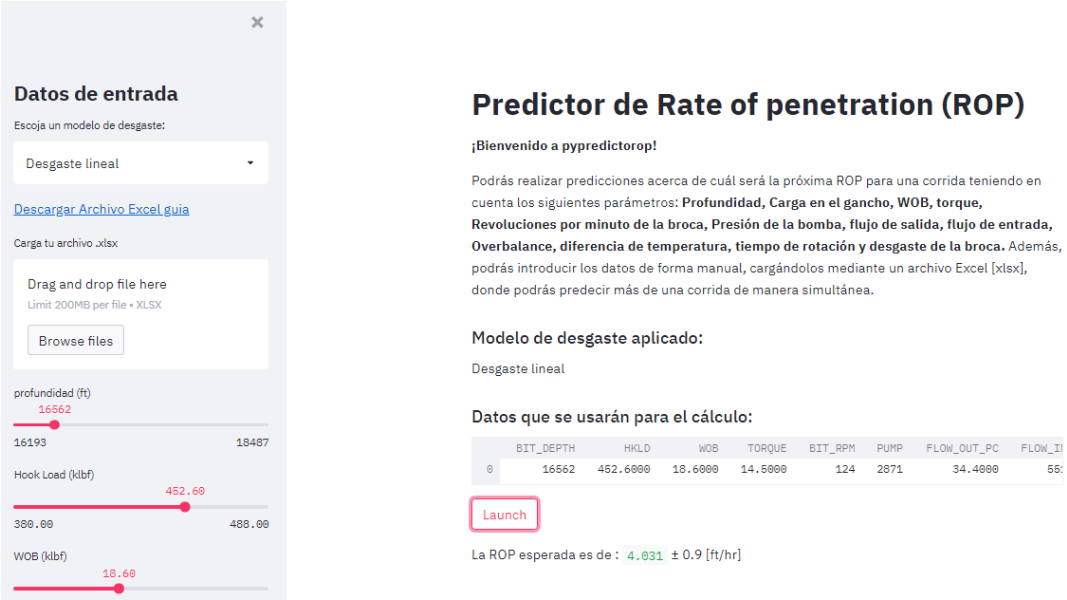
Luego de tener aquella carpeta se procedió a realizar el repositorio en GitHub, el siguiente paso fue realizar el deployment en el servidor externo.

3.5.3. Heroku. Es una plataforma web que se especializa en ofrecer servicios de plataforma administrada, servidores en donde se pueden alojar y desarrollar aplicaciones web en diferentes lenguajes de programación.[48]

Se eligió esta plataforma ya que es de uso gratuito, además se ubica como una de las plataformas de despliegue para startups o empresas pequeñas, en este caso un proyecto de grado. Sin embargo, esta contiene límites que condicionan su uso, uno de ellos es que el slug size durante el desarrollo del microservicio o aplicación web con todos sus parámetros debe pesar menos de 500 MB luego de su compresión, este límite condicionó la posible implementación del modelo extreme gradient boosting, el cual a pesar de ser el modelo con mayor eficiencia, también es el modelo más robusto y pesado, por tanto se decidió realizar un modelado de light gradient boosting machine, ya que luego de su compresión se alcanzó un tamaño de archivo de 356 MB, incluyendo el desgaste lineal y radical.

[48] URRUTIA, Victor.¿Qué es heroku? ¿Para qué sirve?, Ventajas y desventajas. VidelCloud [Online]. 2018.

Figura 34. Interfaz de escritorio aplicación Web

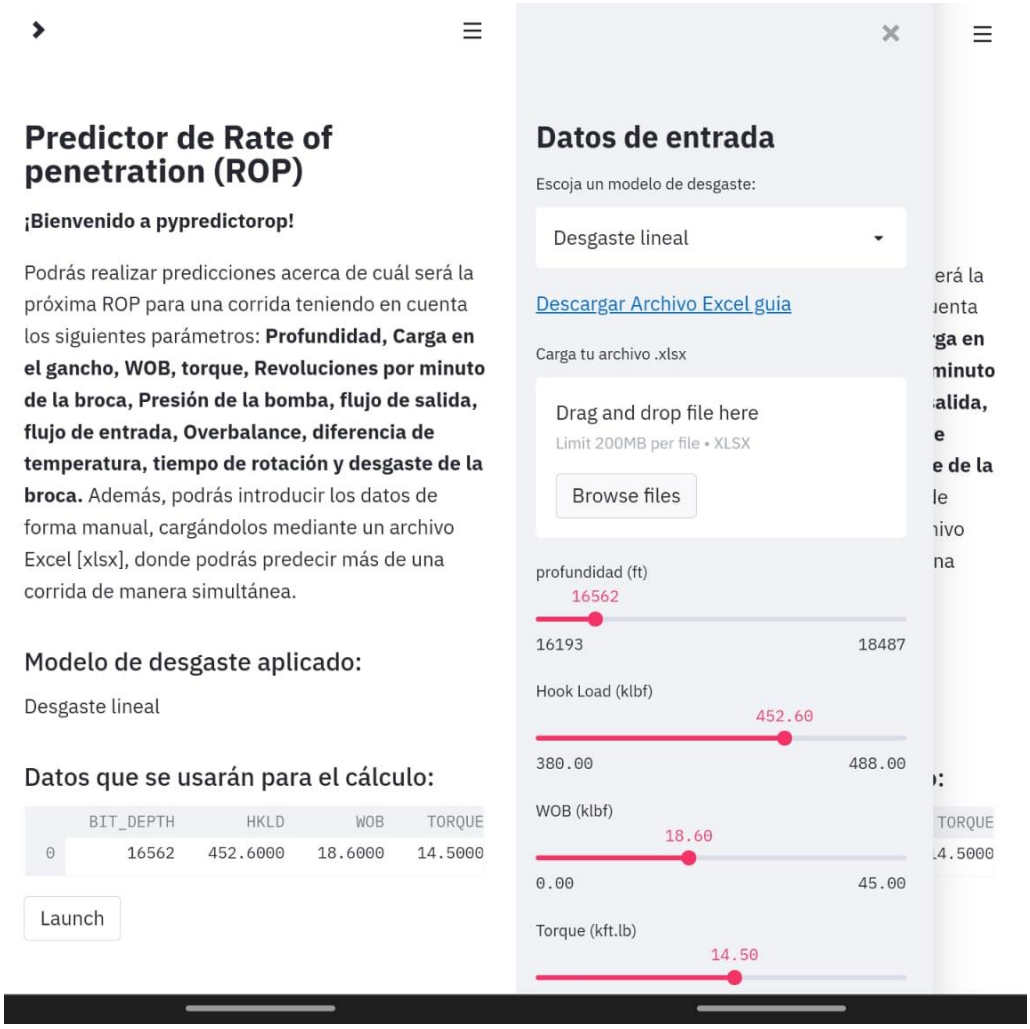


Luego de tener el repositorio preparado y cargado con los archivos especificados anteriormente, se procedió a conectarlo con heroku, mediante la creación de una cuenta gratuita en la plataforma, se enlazó con el repositorio y al realizar el despliegue, antes de esto se le puso un nombre al proyecto que se presentará en la URL de la aplicación y este es “pypredictorop” ya que es un predictor de ROP desarrollado en Python.

Finalmente se obtuvo la URL principal para poder acceder (https://pypredictorop.herokuapp.com), a este servicio se puede ingresar desde cualquier dispositivo, y es multiplataforma, ya que la aplicación web se adapta a teléfonos móviles y versiones escritorio (figuras 34 y 35).

Tras obtener el resultado final, y cumpliendo con los requisitos especificados de dar desarrollo a una aplicación web, usando los modelos de desgaste lineal y radical, haciendo de este servicio una herramienta fundamental para determinar y predecir la ROP en formaciones con características similares siendo formaciones duras y abrasivas, se da por cumplido el objetivo final del proyecto.

Figura 35. Interfaz móvil aplicación web



4. CONCLUSIONES

Los modelos de machine learning generan mejores predicciones que los tradicionales ya que como se evidenció en la literatura, consideran los parámetros de perforación con mayor impacto en la ROP, sin disminuir su desempeño en entornos de perforación complicados como en formaciones duras y abrasivas.

Los métodos de ensemble seleccionados muestran mejores resultados que el modelo KNN, ya que el XGBoost obtuvo mejor desempeño en la predicción de la ROP con un coeficiente de determinación del 74% debido a que es un modelo más robusto y completo, seguido por los modelos LightGBM y KNN con un coeficiente de determinación del 66% y 60% respectivamente.

Al asumir que el comportamiento del desgaste de la broca tiene una tendencia lineal o radical, se observó que los dos tipos de tendencia presentan resultados similares en la capacidad predictiva de los modelos, el uso de un dataset con un tipo de desgaste o el otro genera un aumento en el coeficiente de determinación menor al 0.4%, evidenciando un cambio poco significativo.

El criterio de éxito del 70% en la predicción de la ROP propuesto como objetivo al iniciar esta investigación, se cumple y sobrepasa la expectativa obteniendo el 77% de coeficiente de determinación en los datos de prueba y el 74% en la validación, con el modelo XGBoost, demostrando así la efectividad de los modelos de aprendizaje supervisado de machine learning.

Los modelos implementados presentan mejor rendimiento en la predicción de tasas de penetración bajas, ya que la ROP en la formación objeto de estudio no sobrepasa los 57 ft/hr y la mayor cantidad de sus datos se consolidan hasta los 10 ft/hr.

Con el análisis diagnóstico de las variables se evidenció que el torque y el WOB son las variables con mayor incidencia en el algoritmo del modelo LightGBM y el caudal junto con la profundidad son las de menor incidencia, contrario al algoritmo del modelo XGBoost en donde el caudal y RPM en la broca son las variables con mayor impacto y tanto Hook load como el porcentaje de flujo de salida son las de menor importancia en este modelo, esto debido a que el funcionamiento interno de los modelos es distinto en cada algoritmo y prioriza diferentes parámetros.

Como resultado de este proyecto se deja la herramienta de acceso web PYPREDICTOROP que mediante el ingreso de ciertos parámetros y con ayuda de una interfaz gráfica facilita al usuario conocer cuál será la tasa de penetración durante una o varias corridas de manera simultánea y así poder obtener un mejor panorama operacional y evitar el aumento de costos y el tiempo en la perforación.

5. RECOMENDACIONES

Se recomienda en etapas futuras culminar con el proceso de analítica avanzada continuando con la última fase que sería el análisis prescriptivo, en donde se logra la optimización y se puede determinar por ejemplo qué valores de los parámetros obtienen una mayor ROP.

Los modelos Extra trees Regressor, Random Forest y Catboost Regressor se ajustan muy bien a la base de datos según las métricas de cada uno, pero no hubo un enfoque en estos debido a que emplean mayor tiempo de aprendizaje y usan mayor capacidad del procesador. Se recomienda implementar estos modelos para llegar a obtener un rendimiento en la predicción mucho mejor.

Se recomienda realizar el despliegue en otras plataformas como servicio (PaaS) como Microsoft Azure, Amazon Web Services o Google cloud que cuenten con mayor capacidad y soporten el uso de modelos más pesados y con mayor tiempo de aprendizaje.

Usar los modelos de predicción con otras bases de datos de otros pozos y en otras formaciones similares para analizar sus resultados y desempeño.

Se recomienda tomar como referencia este proyecto siguiendo una metodología similar para evaluar ROP en formaciones blandas o intermedias.

Agregar variables categóricas como el fabricante, tamaño y estado de la broca para analizar su impacto en la predicción de la ROP.

Variar las proporciones de la división de los datos comparando el coeficiente de determinación de los modelos al incrementar y al disminuir el porcentaje de los datos de entrenamiento, prueba y validación del modelo.

Se recomienda implementar modelos de deep learning como redes neuronales y realizar la comparación con los resultados obtenidos de los modelos seleccionados de machine learning.

BIBLIOGRAFÍA

AGUILAR, Adán. Un algoritmo inspirado en la naturaleza para resolver problemas de optimización numérica restringida con alta dimensionalidad. Universidad Veracruzana. 2019.

AMAT, Joaquín. Árboles de decisión, random forest, gradient boosting y C5.0. [Online] 2017. https://www.cienciadedatos.net/documentos/33_arboles_de_prediccion_bagging_random_forest_boosting#Introducción

BARBOSA, Luís Felipe, et al. Machine learning methods applied to drilling rate of penetration prediction and optimization - A review. 2019.

CASTILLO, Luciano. Conociendo GITHUB Documentation. 2017.

CHAPARRO, Mauricio. Factores críticos para la implementación de proyectos que utilizan datos masivos Big Data en organizaciones operadoras de la industria del petróleo y gas en Colombia industria del petróleo y gas en Colombia. 2021.

DIAZ, Ignacio, et al. Guía de asociación entre variables (Pearson y Spearman en SPSS). Universidad de Chile. Santiago de Chile. 2014

DOWNEY, A.B. How to Think Like a Computer Scientist. Learning with Python. Green Tea Press. 2008.

ESPINOZA, Javier Jesús. Application of Random Forest and XGBoost algorithms based on a credit card applications database. Scielo. 2020.

ESTANISLAO, Irigoyen. Industria 4.0 y transformación digital en la industria del petróleo y gas: Situación actual. Petrotecnica. 2019.

FAN, Junliang, et. al. Light Gradient Boosting Machine: An efficient soft computing model for estimating daily reference evapotranspiration with local and external meteorological data. ScienceDirect. 2019.

FAYYAD, Usama. et al. Knowledge Discovery and data mining: Towards an Unifying framework. 1996.

GAGO, Ultrario. Uso de algoritmos de aprendizaje automático aplicados a bases de datos genéticos. Universidad Oberta de Catalunya. Cataluña. 2017.

GALAN CORTINA, Victor. Aplicación De La Metodología Crisp-Dm A Un Proyecto De Minería De Datos En El Entorno Universitario. Universidad Carlos III de Madrid. 2015.

HERRERA, Herbert. Ingeniería de perforación de pozos de petróleo y gas. Universidad Politécnica de Madrid. 2020.

HINESTROZA, Denniye. El Machine Learning A Través De Los Tiempos, Y Los Aportes A La Humanidad. Universidad Libre Seccional Pereira. Pereira. 2018.

IARTIFICIAL. Ensembles: voting, bagging, boosting, stacking.[Online] 2019. https://www.iartificial.net/ensembles-voting-bagging-boosting-stacking/

KHANDELWAL, Pranjal. Which algorithm takes the crown. Analytics Vidhya [Online]. 2017. https://www.analyticsvidhya.com/blog/2017/06/which-algorithm-takes-the-crown-light-gbm-vs-xgboost/

LEJARZA, J. y LEGARZA, I. Distribución normal. 2018.

MAISUECHE CUADRADO, Alberto. Utilización del Machine Learning en la industria 4.0. Universidad de Valladolid. 2019.

MONTES, Abraham, CARREÑO, Wilson y GUÍO, Miguel. Aspectos de la perforación de pozos complejos en piedemonte en tiempos de crisis. El reventón energético. 2018.

MORENO, A. A la espera de un Big Bang de datos. [Online]. 2019. https://www.diarioabierto.es/451036/a-la-espera-de-un-big-bang-de-datos

NAVIDI, William. Estadística Para Ingenieros y Científicos. Mc Graw Hill. Colorado School of Mines. 2006.

NEGRÓN, Pablo Andrés. Redes Neuronales Sigmoidal Con Algoritmo Lm Para Pronostico De Tendencia Del Precio De Las Acciones Del IPSA. Pontificia Universidad Católica de Valparaíso. 2014.

PETERSON, Leif. K-nearest neighbor. Scholarpedia. Scholarpedia [Online]. 2019. http://scholarpedia.org/article/K-nearest_neighbor

PYKES, Kurtis. Hyperparameter Optimization. Towards Data Science. 2010.

QUEVEDO, Fernando. Medidas de tendencia central y dispersión. Universidad de Chile. Santiago de Chile. 2011.

RANJAN, G S K et. al. K-Nearest Neighbors and Grid Search CV Based Real Time Fault Monitoring System for Industries. 2019.

RAVIOSA, Alexandro. Inteligencia Artificial, apuesta obligada para la industria del petróleo. Global Energy [Online], 2020.

RODRIGUEZ, Nathalia y TOBAR, Daniel. Implementación De Un Modelo Predictivo De Machine Learning Para La Estimación De Los Parámetros Óptimos De La Rop Y La Mse En La Sección 8½'' Y 12 ¼'' Para Los Pozos Perforados Con Motor De Fondo En El Campo Yarigui – Cantagallo Durante El 2019. 2021.

SHAFIQUE, Umair y QAISER, Haseeb. A Comparative Study of Data Mining Process Models (KDD, CRISP-DM and SEMMA). University of Gujrat. 2014.

SHERIDAN, Robert. Extreme Gradient Boosting as a Method for Quantitative Structure–Activity Relationships. ASCpublication [Onlines]. 2016.

SPEALMAN, Mark, et al. Digital Transformation Initiative Oil and Gas Industry. World Economic Forum. 2017.

SYRUS. Usos de la inteligencia artificial en la industria petrolera. Syrus España [Online]. 2020. https://syrus.es/usos-de-la-inteligencia-artificial-en-la-industria-petrolera-2291.html

UNAM. Aspectos generales relacionados al corte de núcleos. Universidad Nacional Autónoma de México [Online]. http://www.ptolomeo.unam.mx:8080/xmlui/bitstream /handle/ 132.248.52.100/1107/A7.pdf?sequence=7

URRUTIA, Victor.¿Qué es heroku? ¿Para qué sirve?, Ventajas y desventajas. VidelCloud [Online]. 2018. https://videlcloud.wordpress.com/2018/12/22/que-es-heroku-para-que-sirve-ventajas-y-desventajas/

ZAMORANO, Juan. Comparativa Y Análisis De Algoritmos De Aprendizaje Automático Para La Predicción Del Tipo Predominante De Cubierta Arbórea. Universidad Complutense de Madrid. 2018.

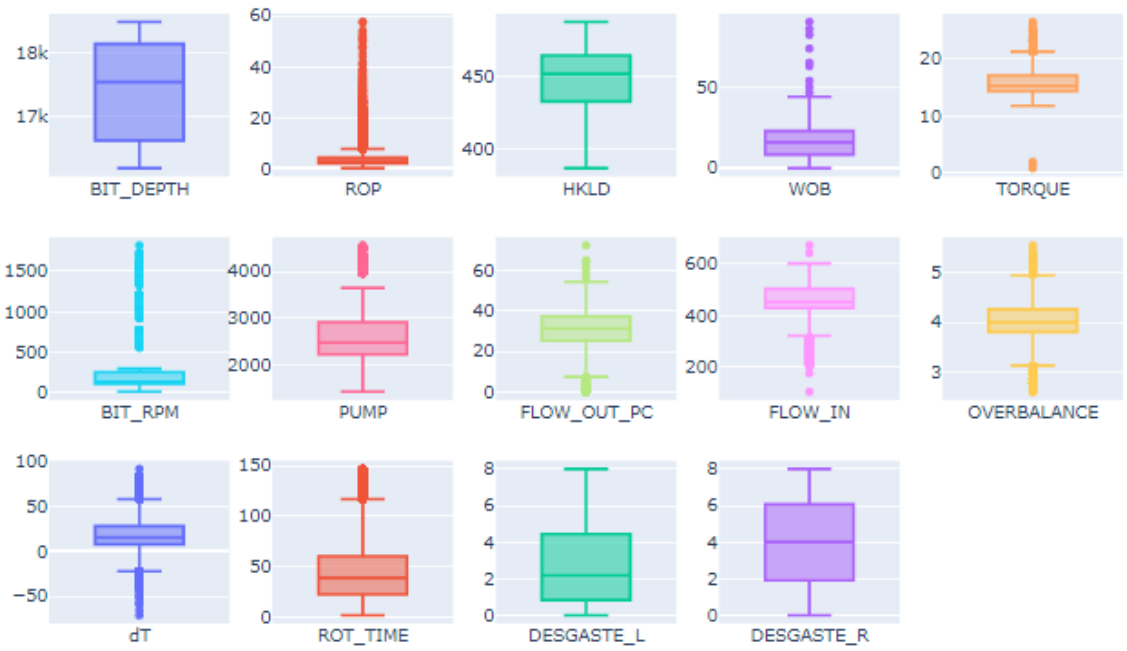
ANEXOS

Anexo A. Tabla de medidas de tendencia central del dataset final

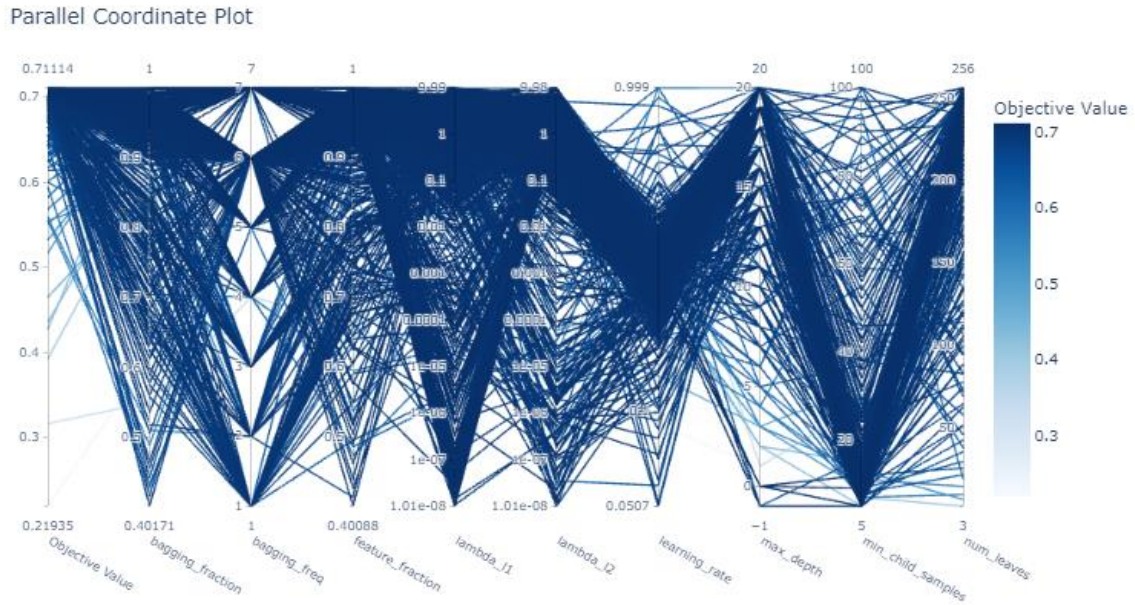
	BIT_DEPTH	ROP	HKLD	WOB	TORQUE	BIT_RPM	PUMP
count	115600.000000	115600.000000	115600.000000	115600.000000	115600.000000	115600.000000	115600.000000
mean	17422.635279	3.546924	449.165922	16.075836	15.513115	345.521817	2652.739723
std	731.874396	2.592602	17.067058	9.259590	2.656833	421.565052	612.895982
min	16193.010000	0.000000	387.220000	0.010000	0.010000	0.000000	1338.000000
25%	16633.442500	2.000000	433.227500	7.880000	14.240000	100.000000	2207.000000
50%	17556.890000	3.000000	451.480000	15.940000	15.260000	126.000000	2455.000000
75%	18132.245000	4.300000	464.270000	23.000000	17.050000	251.000000	2898.000000
max	18487.970000	57.600000	487.320000	90.880000	26.510000	1822.000000	4561.000000

FLOW_OUT_PC	FLOW_IN	OVERBALANCE	dT	ROT_TIME	DESGASTE_L	DESGASTE_R
115600.000000	115600.000000	115600.000000	115600.000000	115600.000000	115600.000000	115600.000000
31.529457	439.692872	3.902689	21.169975	44.540101	2.763566	4.018616
6.826408	94.857721	0.438951	20.394818	27.796547	2.229206	2.307918
0.040000	0.000000	2.620000	-70.900000	0.000000	0.000000	0.000000
25.460000	424.000000	3.800000	8.000000	22.500000	0.847190	1.888131
32.180000	452.000000	4.000000	15.200000	39.100000	2.132840	3.963853
37.550000	503.000000	4.260000	27.600000	60.400000	4.400707	6.036160
72.580000	671.000000	5.530000	91.300000	146.700000	8.000000	8.000000

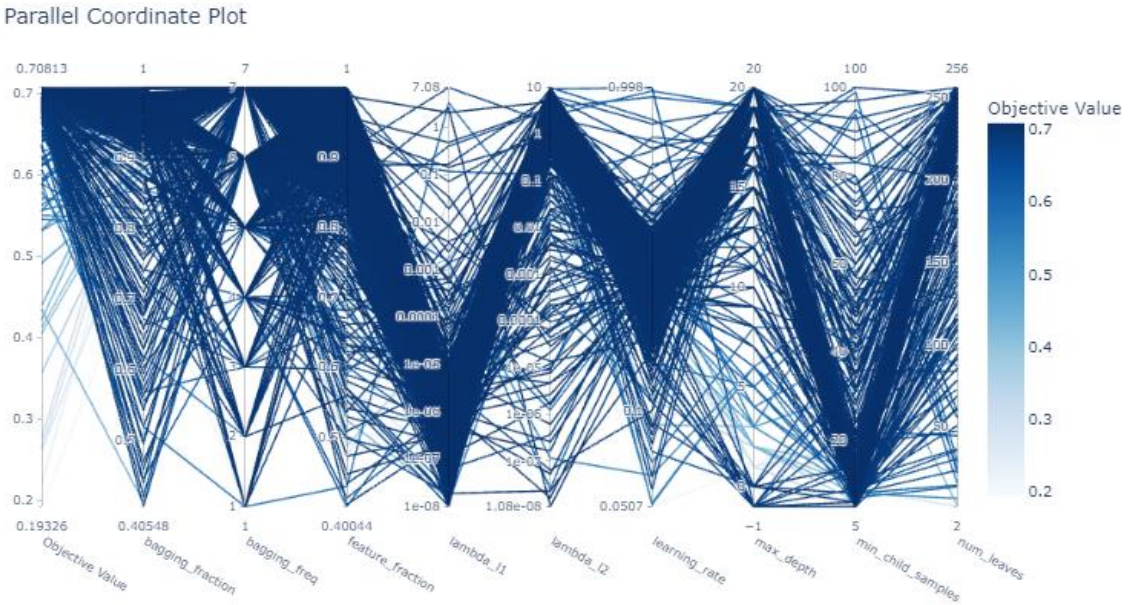
Anexo B. Box plots del dataset final



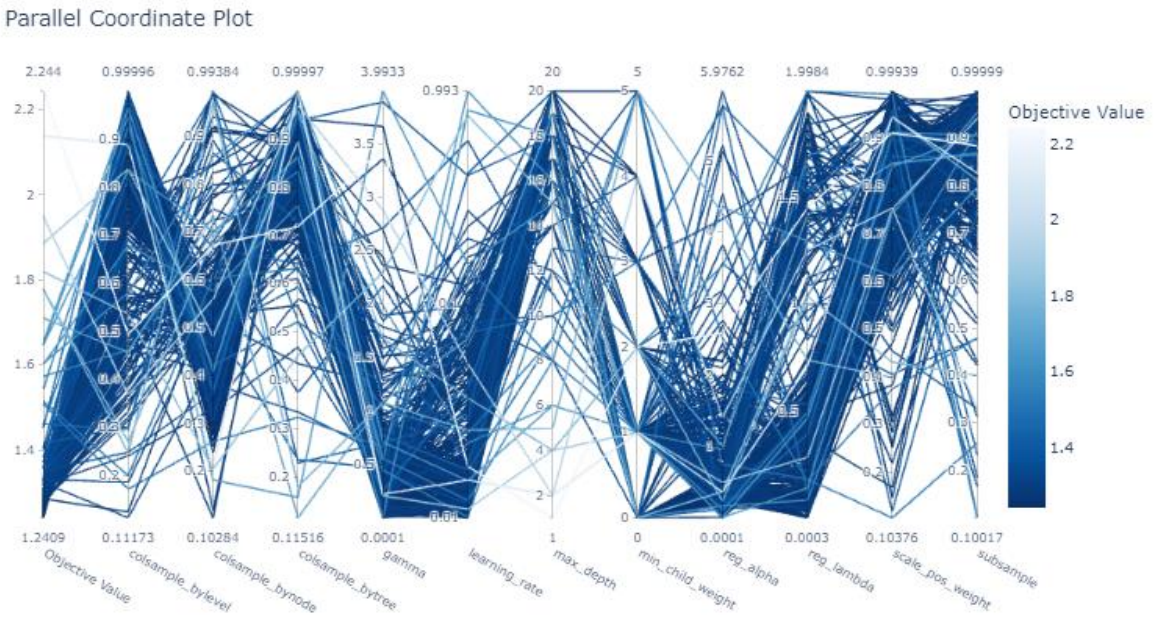
Anexo C. Gráfico de coordenadas paralelas de los parámetros del modelo LightGBM con el dataset D1



Anexo D. Gráfico de coordenadas paralelas de los parámetros del modelo LightGBM con el dataset D2



Anexo E. Gráfico de coordenadas paralelas de los parámetros del modelo XGBoost con el dataset D1



Anexo F. Gráfico de coordenadas paralelas de los parámetros del modelo XGBoost con el dataset D2