ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE SISTEMAS

UNIDAD DE TITULACIÓN

APLICACIÓN DE TÉCNICAS DE MACHINE LEARNING EN LA PREDICCIÓN DE LA CORROSIÓN EN LAS OPERACIONES PETROLERAS DE ECUADOR

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL GRADO DE MAGISTER EN SISTEMAS DE INFORMACIÓN MENCIÓN INTELIGENCIA DE NEGOCIOS Y ANALÍTICA DE DATOS MASIVOS

KLÉVER ALEJANDRO MERA SHUGULI
klever.mera@epn.edu.ec

Director: Henry Patricio Paz Arias
henry.paz@epn.edu.ec

2022

APROBACIÓN DEL DIRECTOR

Como director del trabajo de titulación "*Aplicación de Técnicas de Machine Learning en la Predicción de la Corrosión en las Operaciones Petroleras de Ecuador"* desarrollado por Kléver Alejandro Mera Shuguli, estudiante de la maestría en Sistemas de Información mención Inteligencia de Negocios y Analítica de Datos Masivos, habiendo supervisado la realización de este trabajo y realizado las correcciones correspondientes, doy por aprobada la redacción final del documento escrito para que prosiga con los trámites correspondientes a la sustentación de la Defensa oral.

HENRY PATRICIO PAZ ARIAS

Firmado digitalmente por HENRY PATRICIO PAZ ARIAS
Fecha: 2022.06.30 14:34:17 -05'00'

Henry Patricio Paz Arias

DIRECTOR

DECLARACIÓN DE AUTORÍA

Yo, Kléver Alejandro Mera Shuguli, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Firmado electrónicamente por:
KLEVER
ALEJANDRO MERA
SHUGULI

Kléver Alejandro Mera Shuguli

DEDICATORIA

Dedico este trabajo a mis padres y hermanos con quienes siempre puedo contar en cualquier situación. A mi querida hija Pamela, quien siempre me anima para que continúe con mis estudios y me motiva con su sonrisa. A Mayra, mi amiga y compañera, con quien caminamos juntos en los intrincados caminos de la vida.

AGRADECIMIENTO

Agradezco a Dios y a mis padres por su infinito amor.

A mi director de tesis Henry Paz por su guía y apoyo durante todo el dasarrollo de este trabajo.

ÍNDICE DE CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

LISTA DE ANEXOS

RESUMEN

La presencia de la corrosión en las tuberías de extracción de petróleo no solo genera pérdidas económicas para el Ecuador sino que también tiene efectos nocivos tanto para la salud de las personas como para el medio ambiente. Esta tesis de Maestría propone el uso de algorítmos de Machine Learning para estimar el nivel de corrosión en función de datos recabados respecto a la producción y a los procesos químicos presentes en las líneas de extracción. En este trabajo se implementan los modelos de clasificación Support Vector Machine, Random Forest y XGBoost, los mismos que son evaluados en función de un conjunto de métricas, es decir, se realiza la comparación de los resultados del accuracy, precision, sensitivity, specificity y f1-measure, y según dichos resultados se selecciona un modelo para que posteriormente se realice un ajuste de hiperparámetros para mejorar el performance de este modelo. Además, se considera una etapa de ingeniería de características en la que se incluye la normalización de los datos, la eliminación de las variables con alta correlación y el balanceo del conjunto de datos. También, se incluye una aplicación web tipo dashboard para que el usuario final pueda realizar la predicción de manera amigable. Cabe mencionar que todo este proyecto fue desarrollado utilizado el lenguaje de programación R.

Palabras clave: Corrosión, Machine Learning, Support Vector Machine, Random Forest, XGBoost, Aplicación Web, R.

ABSTRACT

The presence of corrosion in oil extraction pipes not only generates economic losses for Ecuador, but also has harmful effects on both people's health and the environment. This Master's thesis proposes the use of Machine Learning algorithms to estimate the level of corrosion based on data collected regarding production and the chemical processes present in the extraction lines. In this work, the Support Vector Machine, Random Forest and XGBoost classification models are implemented, the same ones that are evaluated based on a set of metrics, that is, the comparison of the results of the accuracy, precision, sensitivity, specificity and f1-measure, and according to these results, a model is selected so that later a hyperparameter tuning is made to improve the performance of this model. In addition, a feature engineering stage is considered, which includes the normalization of the data, the elimination of variables with high correlation and the balancing of the data set. Also, a dashboard-type web application is included so that the end user can make the prediction in a friendly way. It is worth mentioning that this entire project was developed using the R programming language.

Keywords: Corrosion, Machine Learning, Support Vector Machine, Random Forest, XGBoost, Web application, R.

1. INTRODUCCIÓN

Uno de los componentes de mayor relevancia en la economía de Ecuador es el petróleo ya que en promedio, entre los años 2015 y 2020, este producto representó el 32% de todos los bienes exportados y el 28% del total de los ingresos entre el 2018 y el 2020 [1]. Además, según el Banco Central de Ecuador (BCE) los resultados esperados para el 2021 en lo referente a extracción de petróleo, gas natural y actividades de servicio relacionadas, se prevé alcance un 3,32% del total del PIB [2].

El proceso de extracción del petróleo enfrenta varios retos debido a condiciones extremas de temperatura y presión, entre otros factores, que afectan al fluido; estos elementos generan cambios en dicho fluido lo que a su vez puede generar problemas de obstrucción en las tuberías. Entre estos problemas están la corrosión y la incrustación, el primero se refiere al desgaste que sufre la tubería como consecuencia de una reacción química, mientras que el segundo se presenta cuando una costra mineral se impregna en la superficie. En los dos casos, se produce obstrucción que limita el flujo del petróleo en las tuberías [3].

La corrosión, que se mide en milímetros por año (mm/yr), ocasiona varios problemas en las operaciones petroleras como la detención de la producción debido al cierre de líneas hasta que se realicen tareas de mantenimiento y como consecuencia pérdidas económicas. Incluso, los efectos de la corrosión pueden ser muy nocivos tanto para el personal como para el medio ambiente debido a las fugas de hidrocarburos [4].

Por ello, para asegurar el flujo se ejecutan algunas actividades químicas que mitiguen los efectos de este fenómeno como monitorear la corrosión con cupones en superficie, medir el hierro en cada pozo, control y seguimiento de que la inyección del inhibidor de corrosión sea constante y sondas de hidrógeno [5], [6]. Aunque estos métodos ayudan de alguna manera a mitigar los defectos causados por la corrosión, en la práctica no son los más efectivos puesto que en el mejor de los casos se consigue una inhibición máxima esperada de la velocidad de corrosión de 6 mm/yr [7].

Lo anterior, claramente indica que se requieren de otras técnicas para estimar los efectos destructivos causados por la corrosión. En estos casos, el uso de Machine Learning (ML) a través de alguna de las herramientas de aprendizaje supervisado o no supervisado como Random Forest (RF), Neural Networks (NN), Support Vector Machine (SVM), K-means

clustering (KMC), Gradient Boosting Machine (GBM), entre otras, ayudan en la solución de este problema [8], [9], [10].

De acuerdo con el antecedente expuesto, en el presente trabajo se aplicará modelos de ML, como arquitecturas con capacidad de predicción inteligente, con el fin de predecir la corrosión en las tuberías para incrementar la eficiencia, reducir costos y a la vez asegurar el flujo. Para ello, se propone utilizar aprendizaje supervisado y específicamente los modelos SVM, RF y XGBoost para predecir el nivel de corrosión.

1.1. Pregunta de investigación

¿Cómo se pueden aplicar técnicas de ML a las características químicas y productivas de los pozos petroleros para predecir la corrosión en las operaciones petroleras de Ecuador considerando datos obtenidos de los años 2018 al 2020?

1.2. Objetivo general

Aplicar técnicas de ML en la predicción de la corrosión en las operaciones petroleras de Ecuador.

1.3. Objetivos específicos

- Revisar la literatura respecto a la aplicación de ML en la predicción de la corrosión en la industria petrolera.
- Obtener datos referentes a las condiciones de la tubería asociados con la corrosión entre los años 2018 y 2020.
- Realizar un análisis exploratorio y de atributos de los datos recolectados.
- Limpiar y transformar los datos, previo al modelado.
- Seleccionar modelos de ML para predecir la corrosión y las correspondientes métricas de evaluación del modelo como accuracy, precision, sensitivity, specificity, F1 – measure y Area Under the Receiver Operating Characteristics (AUC ROC).
- Evaluar cuáles de las variables son más importantes en la predicción de la corrosión.
- Desarrollar un dashboard de resultados integrado con el mejor modelo de ML mediante visualizaciones y formularios para predecir la corrosión.

1.4. Marco Teórico

1.4.1. Corrosión

En esencia, la corrosión es la degradación de los materiales en ambientes corrosivos que resulta de la reacción simultánea en el ánodo y cátodo de la siguiente manera, la reacción de oxidación se presenta en el ánodo al momento en el que los electrones son emitidos, mientras que la reacción de reducción se lleva a cabo en el cátodo cuando los electrones son aceptados, este proceso continúa hasta el momento en el que se alcanza un estado electroquímicamente estable [11].

Existen varios tipos de corrosión, tales como: a) corrosión uniforme, como su nombre lo dice, produce una pérdida uniforme en la superficie del material; b) corrosión localizada, que remueve el material en un área específica de la superficie expuesta; c) corrosión galvánica, se produce en un medio electrolítico por el contacto entre dos diferentes metales, los metales menos nobles como el zinc, aluminio y magnesio son más susceptibles a este tipo de corrosión; d) corrosión por picadura, es similar a la corrosión localizada pero con la diferencia de que el material removido de pequeñas áreas se debe a que dicha área se vuelve relativamente ánodo respecto al resto del material; e) corrosión por influencia microbiológica, es producida por microorganismos como bacterias, hongos y micro algas; y f) corrosión por tensión, que se presenta como resultado de la combinación de factores metalúrgicos, ambientes corrosivos y estados mecánicos. Los metales son propensos a este tipo de corrosión, la cual inicia como una fisura y luego crece hasta alcanzar un tamaño crítico hasta producir efectos catastróficos [8]. La corrosión genera efectos mecánicos como la cavitación, que produce burbujas de vapor que degradan los elementos de protección de las diferentes partes del sistema, así como también efectos térmicos, que se presentan cuando en las zonas de producción las temperaturas se encuentran entre 0°y 60°.

1.4.1.1. Control de la Corrosión

Respecto al control de la corrosión en pozos terrestres se tienen los siguientes métodos: a) CRA (Corrosion-Resistant Alloy), en el que se considera la utilización de aleaciones resistentes a la corrosión del acero en vez del acero al carbono y de esta manera ofrecer más resistencia a la corrosión pero con la connotación de que el CRA es más caro; b) Inhibidores de corrosión, en este método se usan productos químicos que tienen la finalidad de inhibir la corrosión, los mismos que crean una capa protectora en las paredes de la tubería. Para seleccionar los inhibidores se considera el costo, impacto ambiental y toxicidad y dependiendo del grado de corrugado, así como también de la geometría de la

tubería, se puede alcanzar una eficiencia del inhibidor entre el 85% y el 95%; c) Protección catódica, cuyo objetivo es específicamente la protección contra la corrosión por oxígeno. Esta técnica buscar reducir la diferencia de potencial lo que se consigue aplicando una corriente tanto al ánodo como a la tubería que se requiere proteger, para lo cual se tienen dos opciones que son la protección catódica a través de ánodos galvánico y la protección catódica por corriente impresa; y d) Recubrimientos protectores, para que mediante capas de revestimiento o pintura se proteja la superficie de la tubería de tal manera que se evite el contacto con los fluidos que circulan en dicha tubería, y básicamente están compuesto por elementos no metálicos como el caucho, la fibra de vidrio y epoxi [12].

1.4.1.2. Monitoreo de la corrosión

Con el objetivo de medir el nivel de corrosión presente en las tuberías de los pozos petroleros se utilizan varias técnicas para determinar, ya sea, el espesor de estas o el cambio en la resistencia eléctrica, dicha medición se la realiza en lugares específicos que tienden a presentar desgaste de manera más usual. Para medir el diámetro interno de las tuberías, de tal manera que se detecten agujeros ocasionados por la corrosión, se utiliza el Caliper [13]. Por otro lado, el uso de electrodos en las tuberías a varias alturas permite realizar mediciones de resistencia y diferencia de potencial y de esta manera determinar la tasa de corrosión. Además, con la ayuda de elementos electromagnéticos, que generan campos magnéticos primarios y secundarios, los mismos que al interferirse producen un desplazamiento de fase que es proporcional al espesor de la tubería y así obtener la tasa de corrosión. Finalmente, los elementos ultrasónicos, los cuales emiten pulsos dentro de la tubería y al medir tanto el tiempo de tránsito del primer eco como la frecuencia de la señal resonante se estima el grosor de la tubería y por ende el efecto de la corrosión [13].

1.4.1.3. Predicción de la corrosión

Las técnicas actualmente utilizadas para predecir la corrosión consideran básicamente datos de laboratorio y de campo, con los cuales se realiza la estimación de la tasa de corrosión, y posteriormente se definen las acciones que se ejecutarán enfocadas en prevenir o corregir los efectos causados por la corrosión. Entre los modelos utilizados se encuentran los siguientes [14]: a) Norsok, este modelo es capaz de estimar las tasas de corrosión hasta temperaturas de 150°C y toma en cuenta el efecto de las películas protectoras a altas temperaturas; b) De Waard, es un modelo que considera la correlación entre la temperatura y la presión parcial del óxido de carbono; c) Corplus, el mismo que toma en cuenta el flujo del fluido y el índice corrosivo potencial y expresa la severidad de la corrosión por niveles como alto, bajo, entre otros; y d) Multicorp, modelo que utiliza un

mecanismo de simulación del proceso químico, electroquímico y de transporte que usualmente se presenta en la corrosión.

1.4.2. Machine Learning

Machine Learning (ML), o aprendizaje automático, se refiere al proceso computacional que mediante el uso de datos como entrada consigue realizar una tarea para generar un resultado específico sin que dicha tarea haya sido explícitamente programada. Asimismo, Ethem Alpaydin lo definió como programas para optimizar un criterio usando ejemplos o experiencias pasadas [15]. Uno de los ejemplos más comunes que permite un mejor entendimiento de este concepto es la detección de spam en los correos electrónicos. Desde un punto de vista práctico, se busca crear un filtro que permita diferenciar entre los correos electrónicos que son spam de los que no son, de tal manera que cada vez que se reciba un correo en la bandeja de entrada permanezcan aquellos que no son spam mientras que en la carpeta spam aquellos que efectivamente corresponden a esta categoría [16].

Para lo cual, el proceso inicia con la recolección de datos, en este caso correos electrónicos, los mismos que deben incluir la etiqueta “spam” o “no_spam”, según sea el caso. Luego, el mensaje del correo electrónico debe ser convertido en un vector de características, basado en un diccionario, de tal manera que cada palabra en el mensaje que corresponda con la palabra en el diccionario a la respectiva posición del vector de características se le asigna un valor de 1 caso contrario el valor asignado será de 0. De esta manera se forma el conjunto de datos al cual se le aplica un algoritmo con el fin de obtener un modelo que permita predecir si un correo electrónico o e-mail debe ser clasificado como spam o no [17].

Respecto a la corrosión, gracias al gran crecimiento de la disponibilidad de los datos, se pueden recolectar tanto de las variables asociadas al proceso como de la ocurrencia de la degradación de la tubería y debido a que una de las ventajas de los modelos de ML es que pueden trabajar con múltiples variables, hace que estos algoritmos sean considerados para su aplicación en este tipo de requerimientos, con lo cual se consigue combinar diferentes aspectos y propiedades físico-químicas de las tuberías con las ciencias de la computación y de esta manera obtener productos enfocados en la predicción de la corrosión [18].

Existen cuatro tipos de ML: a) aprendizaje supervisado, b)aprendizaje no supervisado, c) aprendizaje semi-supervisado, y d) aprendizaje por refuerzo.

El conjunto de datos en el aprendizaje supervisado está compuesto por ejemplos etiquetados, dicho conjunto es del tipo $\{(x_i, y_i)\}_{i=1}^{N}$, en el que x_i es conocido como vector de caracteríticas y cada una de sus dimensiones, $x^{(j)}$, contiene un valor que describe de

al ejemplo etiquetado. La etiqueta y_i puede ser de diferente naturaleza, ya sea un elemento de un conjunto fininito de categorías, un número real, un vector, una matriz, entre otros [16]. El ejemplo de la detección de spam en e-mails es del tipo aprendizaje supervisado, en el que cada elemento x representa un e-mail y la etiqueta y_i es "spam" o "no_spam". El aprendizaje supervisado utiliza el conjunto de datos para generar un modelo que reciba como entrada el vector de características x, y entregue como resultado la etiqueta correspondiente a dicho vector.

Para el caso del aprendizaje no supervisado, el conjunto de datos está formado por ejemplos no etiquetados $\{x_i\}_{i=1}^{N}$. De la misma manera que en el aprendizaje supervisado, el vector de características es representado por x. Este tipo de aprendizaje se utiliza en problemas de agrupamiento, en el que a cada vector de características se le asigna un identificador del grupo al que pertenece, es decir, el modelo toma un vector de características x como entrada y luego este vector puede ser transformado en un valor o en otro vector [15].

En lo que tiene que ver con el aprendizaje semi-supervisado, el conjunto de datos incluye muestras tanto etiquetadas como no etiquetadas aunque el número de muestra no etiquetadas es mayor que las etiquetadas. El objetivo de este tipo de aprendizaje es generar un mejor modelo en función de la cantidad de ejemplos no etiquetados [17]. Por ejemplo, los call centers graban las conversaciones con los clientes las mismas que para no ser etiquetadas una a una y utilizar aprendizaje supervisado, se aplica directamente aprendizaje semi-supervisado con el objetivo de inferir características de los clientes.

Finalmente, en el aprendizaje por refuerzo el vector de características se forma en función de la capacidad de la máquina de percibir el estado del medioambiente, dicha máquina o agente puede realizar acciones en cada estado y por cada acción el agente recibe una recompensa. En este aprendizaje, el algoritmo busca aprender una política, que básicamente es una función, que recibe como entrada el vector de características de un estado y entrega como resultado una acción óptima a realizarse en ese estado, entendiéndose por acción óptima a aquella que maximiza la recompensa [15].

1.4.3. Algorítmos de clasificación

1.4.3.1. Support Vector Machine

Support Vector Machine (SVM) es un algortimo de aprendizaje supervisado que se utiliza en problemas de clasificación lineal y no lineal, regresión y detección de outliers o fuera de serie. Para el caso de clasificación, este algoritmo requiere que las etiquetas sean transformadas a números. Si se considera el ejemplo de la detección de spam en un e-

mail, la clasificación se convierte en un problema binario en el que a la etiqueta se le asigna un valor de +1 cuando el e-mail es del tipo spam y el valor de -1 para los casos en los que el e-mail no es spam. El objetivo es separar las dos clases, positivos de negativos (spam y no spam), para lo cual el algorítmo traza una línea imaginaria (o un hiperplano), esta línea es conocida como límite de decisión y se expresa matemáticamente mediante la ecución $wx - b = 0$, en la que w representa un vector de valores reales y b es un número real. En este punto, el algoritmo debe determinar los valores óptimos de w y b de tal manera que el modelo cumpla la siguiente condición: $f(x) = sign(wx - b)$, y de este modo, cuando el modelo recibe una entrada, la predicción devolverá el signo del resultado, en este caso, +1 para spam y -1 para no spam [19].

Para determinar los valores de w y b se debe satifacer la siguiente restricción:

$$\begin{aligned} wx_i - b &\geq +1 \quad if\ y_i = +1 \\ wx_i - b &\leq -1 \quad if\ y_i = -1 \end{aligned}$$

Por otro lado, el margen de separación entre las dos clases, positivas y negativas, debe considerar la máxima distancia, dicha distancia se mide entre los puntos más cercanos de las dos clases hacia el límite de separación, de tal manera de minimizar ‖w‖.

$$\|\mathrm{w}\| = \sqrt{\sum_{j=1}^{D} (w^{(j)})^2}$$

Lo anteriormente indicado se puede aprecia en la figura 1, en que las clases positivas y negativas corresponden a los puntos azules y naranjas respectivamente y la línea roja es el límite de decisión.

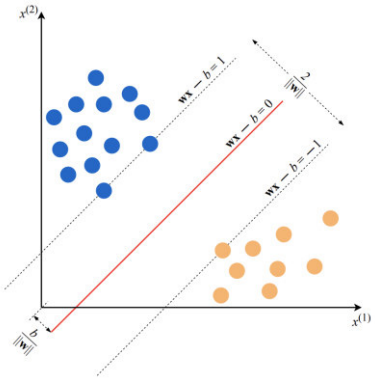

Figura 1 – SVM en dos dimensiones
(Burkov, 2019, pág. 6)

El caso anterior se aplica cuando las clases pueden ser separadas por una línea; sin embargo, cuando no es posible hacerlo con este método, se puede incluir varias funciones polinómicas lo cual podría ralentizar el modelo, es por esto que se crearos los kernels, los que básicamente en vez de incluir varias funciones polinómicas se puede obtener el mismo resultado, como si se las hubiera considerado, pero sin incluirlas [20].

1.4.3.2. Random Forest

Este algoritmo de aprendizaje supervisado se basa en el principio "divide y vencerás", puesto que divide el conjunto de datos en varios subconjuntos, aplica un modelo de predicción a cada subconjunto y luego agrega a todos los predictores. Si el problema es de regresión, la predicción final se obtiene del promedio de los predictores y para el caso de clasificación se considera la clase más votada. En general, el algoritmo parte de un conjunto de datos de entrenamiento del cual se crean B muestras aleatorias (muestreo con reemplazo) y para cada una de dichas muestras se construye un árbol de decisión f_b, con lo cual se consiguen B árboles de decisión y luego, para el caso de regresión, se obtiene el promedio de todos los predictores [21].

$$y \leftarrow \hat{f}(x) = \frac{1}{B}\sum_{b=1}^{B} f_b(x)$$

Random Forest es un algoritmo que puede ser aplicado a una gran variedad de problemas, sean estos de clasificación o regresión, debido a su adaptabilidad a conjuntos de datos de alta dimensión y muestras de tamaños pequeños, así como también su capacidad de trabajar en paralelo ya sea en múltiples CPUs o en diferentes servidores. Por otro lado, este algoritmo es un tipo de lo que se conoce como ensemble learning o aprendizaje conjunto, el mismo que está formado por un grupo de predictores (de clasificación o regresión), estos predictores realizarán la predicción individualmente y luego en función de la clase más votada, o el promedio, se define la predicción resultante del ensamble. Además, para evitar la correlación entre los árboles este modelo realiza una inspección de un conjunto aleatorio de las variables predictoras en cada división, puesto que, la correlación hará que los malos modelos tengan mayor probabilidad de estar de acuerdo lo que a su vez afectará ya sea el voto mayoritario para el caso de la clasificación o el promedio en los problemas de regresión, y puesto que los predictores correlacionados no ayudan a mejor la precisión de la predicción, es preferible evitar la correlación de los árboles [22].

Respecto al ajuste de hiperparámetros, Random Forest considera tanto los parámetros de control de crecimiento de los árboles, así como también los referentes al control del ensamble. En este contexto, los parámetros más importantes a ser considerados para su ajuste incluyen el número de árboles y el tamaño del subconjunto aleatorio de variables predictoras a ser tomadas en cuenta en cada división. Este modelo es uno de los más utilizados en proyectos de ML debido a que al considerar múltiples muestras del conjunto de datos original, por un lado, se reduce la varianza del modelo final y por lo tanto el overfitting, que básicamente se refiere a cuando un modelo trata de ajustarse a las pequeñas variaciones del conjunto de datos debido a que el mismo es una reducida muestra de la población; y por otro lado, también se reducen los efectos del ruido, puntos fuera de serie y ejemplos de datos sub y sobre representados [21].

1.4.3.3. XGBoost

XGBoost o eXtreme Gradient Boost es un algoritmo ensamblado con boosting a diferencia de Random Forest que utiliza bagging. Boosting consiste en tomar el conjunto de datos original y de manera secuencial modelarlo usando múltiples modelos del tipo aprendices débiles (waek learners), el resultado final se obtiene al combinar todos los aprendices débiles que fueron generados. Por ejemplo, el conjunto de datos de entrenamiento va a ser modelado por el primer modelo, luego en función de los resultados obtenidos por el primer modelo el segundo modelo intenta corregir los errores del primero; a continuación, el segundo modelo enviará sus resultados al tercer modelo para que este a su vez intente corregir los errores del segundo y así sucesivamente hasta completar la cantidad de aprendices débiles definidos y finalmente se combinan los resultados de todos los modelos para crear un aprendiz fuerte de tal manera de obtener una mejor predicción [23].

Este algoritmo de aprendizaje supervisado ha demostrado ser eficiente en su rendimiento tanto en competencias como en problemas reales de ML, como clasificación de eventos de alta energía física o predicción de comportamiento del consumidor, lo cual se debe a que es paralelizable con lo que gana rapidez en el proceso de aprendizaje ya que los árboles son creados en múltiples núcleos, así como también a su capacidad de escalabilidad puesto que puede manejar billones de ejemplos en configuraciones distribuidas y de memoria limitada.

Respecto al sustento matemático de XGBoost, básicamente se fundamenta en la aproximación de funciones mediante la optimización de funciones de pérdida específicas en conjunto con la aplicación de técnicas de regularización. En este contexto, la función objetivo que se requiere minimizar es $\mathcal{L}^{(t)}$, y en la misma se aprecia que la función l es una suma del anterior y el actual árbol.

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t)$$

En esta ecuación, con el objetivo de utilizar las técnicas tradicionales de optimización, se requiere transformar la función objetivo inicial a una función en el dominio euclidiano, para lo cual se considera la función de aproximación de Taylor $f(a) + f'(a)(x - a)$, de tal manera que se obtiene una nueva función objetivo simplificada [24]:

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^{n} [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t)$$

A esta última función se debe considerar las condiciones que permitan minimizarla, las cuales son:

$$argmin_x \;\; Gx + \frac{1}{2} Hx^2 = -\frac{G}{H}, H > 0$$

$$min_x \;\; Gx + \frac{1}{2} Hx^2 = -\frac{1}{2}\frac{G^2}{H}$$

2. METODOLOGÍA

Para este proyecto se considera la aplicación de la metodología Cross Industry Standard Process for Data Mining (CRISP-DM) ya que la misma permite: a) traducir problemas o necesidades en tareas de minería de datos, b) entender, preparar, transformar y modelar los datos, c) evaluar la efectividad de los resultados, y d) documentar el proceso [25]. CRISP-DM está dividida en seis fases: a) Entendimiento del problema, en esta fase se definen los objetivos y criterios de éxito, se realiza un análisis de la situación actual de estos elementos así como también los objetivos y criterios de éxito de la analítica de datos; b) Entendimiento de los datos, junto con la recolección de los datos, se hace un análisis exploratorio de estos y se valida la calidad de dichos datos; c) Preparación de los datos, se describen los datos, y se realizan tareas de selección, limpieza, construcción, integración y formateo de los mismos; d) Modelamiento, se selecciona un modelo adecuado para responder al problema, se prueba y se miden los resultados del modelo; e) Evaluación, los resultados obtenidos se contrastan contra los objetivos y criterios de éxito y se genera una lista de posibles acciones y decisiones; y f) Despliegue, se elaboran planes de desarrollo, monitoreo y mantenimiento del producto final así como también los reportes y documentación [26].

2.1. Entendimiento del problema

Como se indicó en párrafos anteriores, la corrosión genera efectos nocivos en las líneas de tuberías de extracción de petróleo, lo que a su vez ocasiona pérdidas económicas cuantiosas debido a que la producción debe detenerse hasta que se ejecuten las actividades correctivas necesarias en función de los daños causados por la corrosión. Por otro lado, las técnicas actuales de predicción de la corrosión, indicadas en el punt 1.4.1.3, resultan insuficientes puesto que las mismas no han sido del todo efectivas. Es por esto que, el objetivo principal es predecir la corrosión con mayor precisión de tal manera que se puedan tomar decisiones y acciones preventivas a tiempo antes de que los efectos de la corrosión ocasionen los problemas anteriormente mencionados.

En este contexto, se propone el uso de ML como herramienta predictiva que permita cumplir con el objetivo. Dicha propuesta se sustente en el hecho de que en otros mercados justamente se han aplicado estas herramientas obteniendo resultados más precisos [8], [10].

2.2. Entendimiento de los datos

Para el presente trabajo se utilizarán datos históricos de los años 2018 al 2020, y debido a la confidencialidad de la información la fuente de los datos no será revelada. Respecto a las variables a ser consideradas para la predicción de la corrosión, se tomaron en cuenta tanto las relacionadas con los procesos químicos así como también las de producción (Tabla 1).

Tabla 1 – Variables de entrada utilizadas para predecir la corrosión

Tipo	Variable	Descripción	Unidad
Producción	bapd bppd bsw mscf	Caudal de agua Caudal de petróleo Sedimento básico y agua Caudal de gas	barriles/día barriles/día % mscfd
Química	pco2 temp Na pH SO4 Cl Fe bicarb Press Ca	Presión parcial CO2 Temperatura de cabeza Sodio pH Concentración de sulfatos en agua Concentración de cloruros en agua Contenido de hierro Bicarbonatos Presión Concentración de calcio en agua	Psi °F ppm Na - ppm SO4 ppm Cl ppm Fe ppm HCO3 Psi ppm Ca

En lo que tiene que ver con la variable a predecir, mpy, la Tabla 2 presenta la información respectiva.

Tabla 2 – Variable a predecir

Tipo	Variable	Descripción	Rango	Criticidad
Producción	mpy	Velocidad de corrosión	< 1 1 a 4.9 5 a 10 > 10	BAJO MEDIO ALTO SEVERO

2.2.1. Recolección de datos

Una vez definidas las variables, el siguiente paso es recolectarlas, para lo cual se dispuso de varios archivos en formato Excel en los cuales dichas variables se encontraban

disgregadas en diferentes formatos y también con diferente disposición ya que en unos casos las variables se encuentran registradas en columnas y en otros casos en filas. Además, para un grupo de variables, los datos se encontraban en un solo archivo mientras que en otros casos los datos estaban divididos por mes por lo que se tenían hasta un total de 36 archivos [27].

Para la extracción de los datos se definieron las secuecias por variable y fuente, es así que para el caso de las variables bapd, bppd y bsw, dado que los datos tienen una frecuencia diaria y están dispuestos en filas, luego de extraerlos del respectivo archivo de Excel, era necesario transformar de filas a columnas con lo cual la primera columna es el nombre del pozo, la segunda la fecha y la tercera columna el valor de la variable. En el caso de la variable mscf, los datos, que también son diarios, están divididos en archivos mensuales por lo que se generó el código para leer cada archivo, ordenar las columnas para que la disposición sea la misma que para la variable bapd, es decir, pozo, fecha y valor, y luego combinar todos los conjuntos de datos parciales de cada mes en un solo conjunto. Se tuvo que leer cada archivo independientemente debido a que cada archivo tenía los datos en diferente fila e incluso, en algunos casos, en diferentes columnas.

Respecto a la variable pco2, los datos no son diarios sino mensuales, en cada columna del archivo existe una etiqueta que indica el mes y año, por ejemplo, 'Apr-18', por lo que luego de extraerlos se tuvo que formatear la fecha y seguidamente, al igual que en los casos anteriores disponer los datos en las tres columnas indicadas. Finalmente, para las variables temp, Na, pH, SO4, Cl, Fe, bicarb, Press y Ca, las mismas se encuentran en un solo archivo en el que las filas indican la fecha y el nombre de la variable, las columnas son los valores correspondientes y cada pestaña del archivo representa a un pozo específico. Estos datos tampoco son diarios, tienen frecuencias diferentes (trimestral, semestral y esporádica). Luego de extraerlos, se forma un solo conjunto de datos con la misma disposición de pozo, fecha y valor.

Para el caso de la variable mpy, los datos fueron extraídos siguiendo el mismo procedimiento que para la variable bapd dado que estos datos también son diarios, aunque con un reducido número de valores válidos. Luego de extraerlos, en función de los valores se definieron los rangos para crear la etiqueta correspondiente al nivel de criticidad indicado en la Tabla 2.

2.2.2. Descripción de datos

Luego de extraer todos los datos y disponerlos de manera estructurada se obtuvieron conjuntos de datos con diferente número de registros debido a que los datos de la diferentes variables no tienen todos una misma periodicidad en el sentido de que algunas

variables son diarias mientra que otras son mensuales, trimestrales, semestrales e incluso esporádicas. Es por esto que en la etapa de preparación de datos, se consideró cada una de las particularidades de las variables de tal manera de generar un solo conjunto de datos que incluya todas la variables tanto las predictoras como la que se busca predecir.

2.3. Preparación de los datos

De acuerdo con la información de la Tabla 1, se definieron 14 variables predictoras o independientes y una variable dependiente que es la que se busca predecir. Debido a la procedencia así como también a las diferentes estructuras de las que provienen los datos se deben realizar procedimientos de limpieza que incluye el cambio de tipo de variable, tratamiento de valores fuera de serie o outliers, imputación de datos para valores nulos y aumento de datos. El objetivo es que el nuevo conjunto de datos cumpla con los requisitos de lo que se conoce como tidy data, en la que cada columna representa una variable, cada fila una observación y cada tipo de unidad de observación es una tabla [28].

2.3.1. Limpieza de datos

Al momento de extraer los datos desde los archivos de Excel, los mismos fueron leídos como caracteres, por lo que primero fue necesario convertirlos a variables de tipo numéricas. Luego, se eliminaros las filas que contenía solo valores nulos o que no contenían ningún valor. Respecto a los valores fuera de serie, se consideraron solamente aquellos que se encontraban dentro del rango definido para cada variable (Tabla 3) y para la imputación de datos se consideró la mediana [29]. Finalmente se eliminaros las filas que contenían valores duplicados. En lo que tiene que ver con las fechas, en los archivos fuente de Excel dicha fecha en algunos casos estaba con formato numérico por lo que también fue necesario realizar una transformación hacia el tipo fecha [30].

Además, para el etiquetado de la variable a predecir, criticidad, en función de los valores de la variable mpy y según los rangos indicados en la tabla 2 se crearon las etiquetas de la siguiente manera: si el valor de mpy es menor a 1 la etiqueta es BAJO, si está entre 1 y 4.9 la etiqueta es MEDIO, para los valores entre 5 y 10 se le asigna la etiqueta ALTO y finalmente si el valor de mpy es mayor a 10 entonces la etiqueta es SEVERO.

Tabla 3 – Rangos de variable independientes

Variable	Rango
bapd	> 0 – 10000
bppd	> 0 – 10000
bsw	> 0 – 100
mscf	> 0 – 10000

Variable	Rango
pco2	> 0 – 500
temp	> 50 – 350
Na	> 0 – 10000
pH	3. 5 – 6.5
SO4	-
Cl	> 0 – 200000
Fe	> 0 – 1000
bicarb	-
Press	> 0 – 500
Ca	> 0 – 50000
mpy	0 -100

Una vez realizado el proceso de limpieza indicado anteriormente, cada variable consta de tres campos, que son: pozo, fecha y valor, y dado que cada variable tiene su propia periodicidad, los conjuntos de datos inicialmente tienen tamaños diferentes como se aprecia en la Tabla 4.

Tabla 4 – Número de registros de cada variable

Variable	Tamaño
bapd	213 745
bppd	215 638
bsw	214 824
mscf	15 948
pco2	6 578
temp	222
Na	222
pH	222
SO4	222
Cl	222
Fe	222
bicarb	222
Press	222
Ca	222
mpy	136 144

Para el caso de las variables con un número reducido de observaciones, lo que se debe básicamente a las diferencias en la periodicidad de los datos, fue necesario ampliar el conjunto de datos de tal manera que todos los conjuntos de datos tengan una misma periodicidad. Para este fin lo que se hizo fue agrupar los datos por pozo y luego completarlos para las fechas faltantes teniendo en cuenta la fecha máxima y mínima de cada conjunto de datos y que el paso entre cada fecha sea por día. La forma como se

completaron los datos fue considerando la condición abajo-arriba de tal manera que cada vez que el algoritmo encontraba un campo vacío leía primero hacia abajo, si encontraba un dato lo tomaba caso contrario leía hacia arriba y toma el dato respectivo [31]. La Tabla 5 muestra el tamaño final de las variables que requirieron el aumento de datos.

Tabla 5 – Número de registros de cada variable

Variable	Tamaño
mscf	163 700
pco2	194 768
temp	12 189
Na	12 189
pH	12 189
SO4	12 189
Cl	12 189
Fe	12 189
bicarb	12 189
Press	12 189
Ca	12 189

A continuación, para formar el conjunto de datos que agrupe todas las variables, independientes y dependientes fue necesario realizar un 'join' de los diferentes conjuntos de datos para lo cual se consideraron pozo y fecha como variables comunes de todos los conjuntos de datos [32].

2.3.2. Análisis exploratorio de datos

El conjunto de datos resultante del proceso de limpieza y preparación está formado por 9212 registros y 18 variables, la Tabla 6 presenta las variables y el tipo.

Tabla 6 – Tipo de variables

Variable	Tipo
Pozo	<chr>
date	<date>
bapd	<dbl>
bppd	<dbl>
bsw	<dbl>
mscf	<dbl>
pco2	<dbl>
temp	<dbl>
Na	<dbl>
pH	<dbl>
SO4	<dbl>

Variable	Tipo
Cl	<dbl>
Fe	<dbl>
bicarb	<dbl>
Press	<dbl>
Ca	<dbl>
mpy	<dbl>
criticidad	<fct>

La Tabla 7 muestra el resumen estadístico de cada una de las variables luego de terminar con el proceso de preparación de los datos.

Tabla 7 – Resumen estadístico de las variables independientes y dependiente

Variable	Mínimo	Q1	Mediana	Promedio	Q3	Máximo
bapd	62.92	563.5	1173.0	1792.0	3119.8	6282.0
bppd	0.041	153.28	249.6	317.37	383.25	1351.6
bsw	20.0	68.0	87.0	77.23	92.0	100.0
mscf	6.15	43.0	78.0	108.89	162.0	2240.1
pco2	1.25	7.68	10.92	19.45	18.72	165.0
temp	95.0	163.0	182.0	183.7	210.0	268.0
Na	2989	23844	29725	28303	32843	53246
pH	4.885	6.1	6.295	6.269	6.448	6.5
SO4	1.0	45.0	95.0	95.63	125.0	550.0
Cl	4500	47500	59700	54562	64600	98000
Fe	1.0	33.0	51.25	51.75	64.64	172.1
bicarb	40.0	180.0	250.0	342.4	341.6	1805.6
Press	50.0	155.0	180.0	167.9	200.0	460.0
Ca	208	4600	5480	4919	6220	10000
mpy	0.176	1.1428	1.8536	4.3428	3.3436	74.333

En la Figura 2 se aprecia la matriz de las distribuciones, correlaciones y gráficas de puntos de las variables numéricas del conjunto de datos final. En la diagonal principal se muestra la distribución de cada variable, en la parte superior las correlaciones y en la parte inferior de la matriz el gráfico de puntos de cada par de variables. Por ejemplo, la correlación entre las variables bapd y bppd es de 0.161, en la esquina superior izquierda está la distribución de la variable bapd y bajo esta última está el gráfico de puntos de las variables bapd vs bppd. Además, se aprecia una correlación fuerte, de 0.991, entre las variables Cl y Na, luego está la correlación entre Ca y Cl (0.881) y en tercer lugar Ca con Na con un valor de correlación de 0.825 [33].

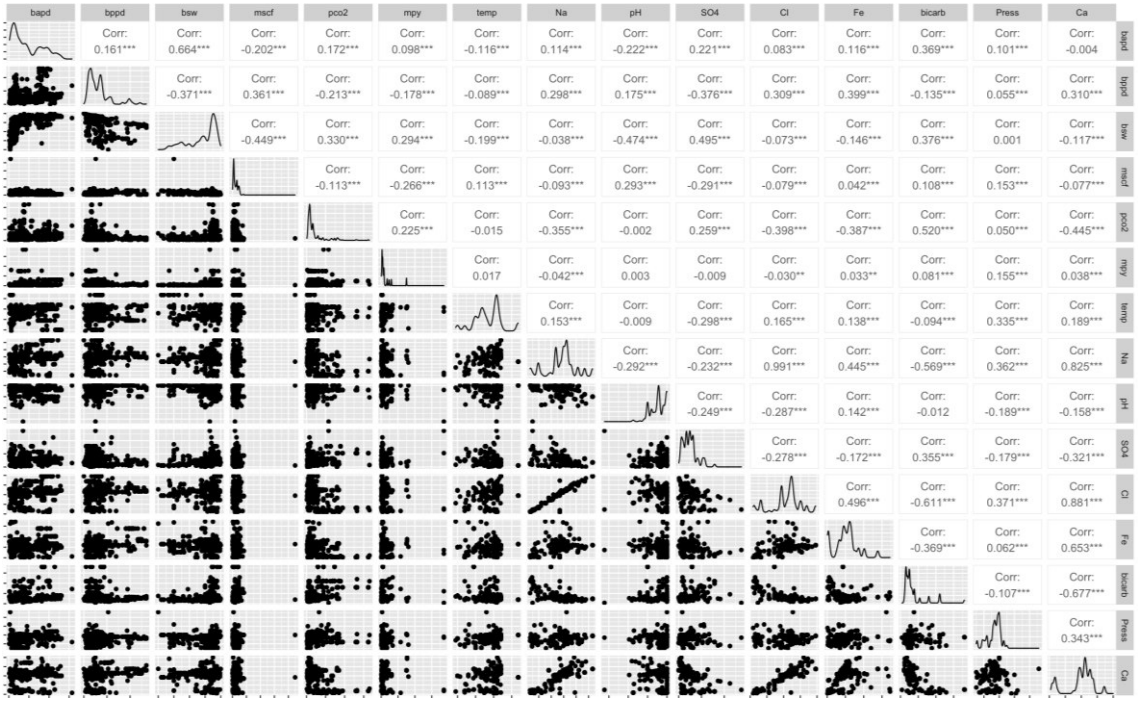

Figura 2 – Distribución y correlación de las variables numéricas

2.4. Modelamiento

Para este proyecto, se utilizaron tres modelos de clasificación con el fin de predecir los niveles de criticidad de la corrosión que fueron indicados en la Tabla 2. Los modelos seleccionados son: Support Vector Machine (SVM), Random Forest (RF) y Extreme Gradient Boost (XGBoost).

2.4.1. Herramientas

Para la etapa de modelado se utilizó el ecosistema Tidymodels, que es un marco referencial del lenguaje de programación R formado por varias librerías de ML. De todo el conjunto de paquetes de ML disponibles en tidymodels se utilizaron los siguientes: a) rsample, librería cuya función principal es la división de la data en conjuntos de datos de entrenamiento y prueba; b) recipes, es el paquete que permite realizar tareas de ingeniería de características; c) parsnip, se encarga de la parametrización de modelos de ML; d) tune y dials, que son los paquetes responsables del ajuste de hiperparámetros de los modelos; y e) yardstick, que es el paquete que permite definir las métricas de evaluación del performance del modelo [34]. La figura 3 muestra el proceso y los paquetes utilizados en cada etapa.

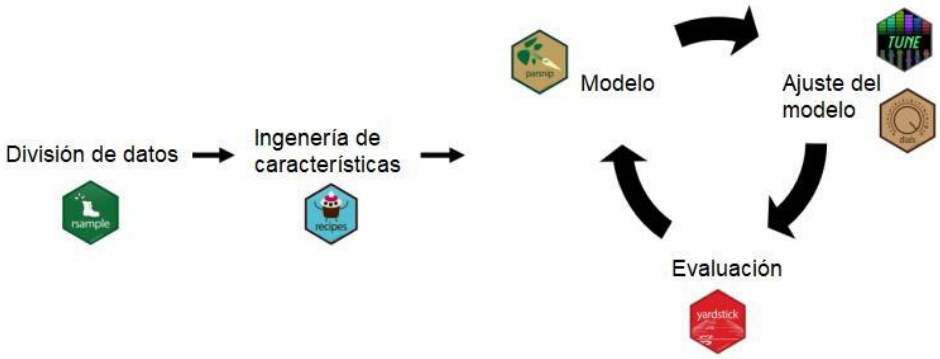

Figura 3 – Librerías y proceso del modelado.
(Svancer, 2021)

2.4.2. Desarrollo del modelado

Una vez definido el conjunto de datos final, así como también las herramientas a ser consideradas para esta etapa, el primer paso es configurar la semilla con el objetivo de tener en cuenta la reproducibilidad [35]. Del conjunto de datos final se eliminaron las variables Pozo y fecha y luego se descartaron las filas duplicadas con lo cual el nuevo conjunto de datos estaba formado por 5891 observaciones y 15 variables. Para la división del conjunto de datos se utilizó el paquete rsample y se consideró 70% para entrenamiento y 30% para prueba [36]. A continuación, se definieron los parámetros para cada modelo, para lo cual, y con el uso del paquete parsnip, se configuró el 'mode' y 'engine' de cada modelo. Respecto al mode, para los tres modelos el parámetro fue 'clasificación' y para el engine, en el caso del modelo SVM, el parámetro se configuró como 'kernlab', para el modelo RF como 'ranger' y finalmente para XGBoost el engine fue 'xgboost' [37].

2.4.3. Ingeniería de características

El siguiente paso es la ingeniería de características [38], proceso en el cual, en esencia, se determinan tanto el número como el tipo de variables a ser consideradas para el modelado de tal manera que se tengan en cuenta aquellas características que aporten al modelo y se eviten errores en la predicción y generalización [39]. En este punto se tomó en cuenta la correlación, que como se comentó respecto a los resultados de la figura 2, en algunas variables dicha correlación supera el valor de 0.9, es por ello por lo que, el primer paso en la ingeniería de características es eliminar la o las variables cuya correlación supere el umbral de 0.9.

A continuación, se considera la normalización, como se aprecia en la tabla 7 existe un amplio rango de valores, así como también diferentes escalas entre las variables por lo que es necesario normalizarlas, este proceso, en este caso, consiste en que todas las variables

tengan desviación estándar de uno y su media sea cero. En los casos de clasificación los modelos determinan la distancia euclidiana entre dos puntos, por lo que si una de las variables del conjunto de datos tiene un amplio rango cuando se calcule la distancia dicho cálculo será mayormente influenciado por esta variable [40].

El tercer paso de la ingeniería de características para este proyecto tiene que ver con el balanceo del conjunto de datos. En los problemas de clasificación no es común que las clases tengan la misma cantidad de muestras, usualmente lo que sucede es que una de las clases tiene un mayor número de ejemplos respecto a las otras, lo cual afecta al rendimiento del modelo puesto que tendrá dificultades para aprender sobre las clases con menor número de muestras [41]. Como se aprecia en la Figura 4, el conjunto de datos es desbalanceado ya que el número de muestra de cada clase es diferente por lo que se ha considerado el balanceo. Para este fin, el método seleccionado fue SMOTE (Synthetic Minority Over-Sampling Technique), el que en términos generales realiza un re-muestreo de la clase minoritaria, para crear muestras sintéticas mediante combinaciones lineales, hasta alcanzar el tamaño de la clase mayoritaria de tal manera que todas las clases tengan el mismo número de ejemplos [42].

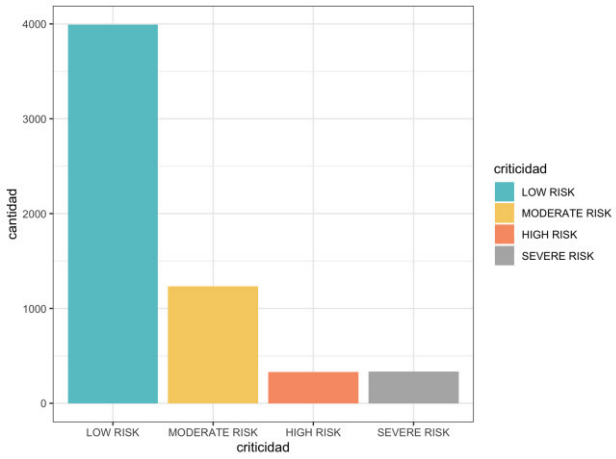

Figura 4 – Distribución de la variable objetivo.

Los pasos indicados anteriormente se consolidan en un recipiente que será aplicado al conjunto de datos de entrenamiento. El paquete utilizado para todas las tareas de ingeniería de características fue parsnip [43]. Adicionalmente, también se aplicó la técnica validación cruzada k-fold (k-fold cross-validation) la misma que consiste en generar varias particiones del conjunto de datos de entrenamiento con el objetivo de que cada partición sea utilizada para varios propósitos de tal manera que se mejore la confiabilidad de los resultados. El número de particiones se define con el criterio k-fold que para este proyecto se consideró k = 10, con lo cual se crearon diez particiones, de las cuales nueve son utilizadas para entrenamiento y una para prueba. Este proceso se repite por diez ocasiones

con la salvedad de que en cada ocasión la partición de prueba es diferente y las restantes nueve particiones se utilizan para entrenamiento [44].

2.4.4. Flujos de trabajo

Cabe mencionar que en el modelado también se utilizó la librería workflows, dicha librería tiene como fin encapsular el preprocesamiento, modelamiento y postprocesamiento. En la etapa de preprocesamiento se incluyen dos elementos: la fórmula del modelo y el recipiente de la ingeniería de características, en este proyecto solo se consideró el segundo elemento y se lo incluyó en el flujo con la instrucción 'add_recipe'. La etapa del modelamiento, en la cual el o los modelos definidos previamente, en este caso SVM, RF y XGBoost, fueron adicionados al flujo de trabajo mediante el objeto 'add_model', y para la tercera etapa, el post-procesamiento, en la cual en próximas versiones del paquete workflows se podrá modificar el umbral de probabilidad para problemas de dos clases, calibrar las estimaciones de probabilidad, truncar el posible rango de predicciones, entre otras funcionalidades [45]. Seguidamente, los flujos de trabajo o workflows fueron entrenados considerando todos los parámetros indicados anteriormente y luego evaluados en función de las métricas definidas para este trabajo.

Respecto a la configuración específica de cada uno de los modelos de clasificación, para SVM se consideró la función básica radial y el motor kernel. Los parámetros de optimización de este modelo son: cost (costo) y rbf_sigma (sigma de la función de base radial), cuyos valores por defecto para cost es 1.0 y para rbf_sigma el mismo no tiene un valor por defecto, sino que en función de los datos se lo estima. En lo referente al modelo RF, la función utilizada es ranger. Este modelo tiene tres parámetros de optimización que son: mtry (predictores seleccionados al azar), que para determinar el valor por defecto considera el valor mínimo entero de la raíz cuadrada del número de columnas del conjunto de datos; trees (número de árboles), cuyo valor por defecto es 500; y min_n (tamaño mínimo de nodo), en el que el valor por defecto para clasificación es 10. Finalmente, para el modelo XGBoost el motor es justamente xgboost y los parámetros de optimización son: mtry (predictores seleccionados al azar), su valor por defecto es el número de todos los predictores; trees (número de árboles), con un valor por defecto de 15; min_n (tamaño mínimo de nodo), 1 como valor por defecto; tree_depth (profundidad del árbol), en el que 6 es el valor por defecto; learn_rate (tasa de aprendizaje) cuyo valor por defecto es 0.3, loss_reduction (reducción de pérdida mínima), 0.0 como valor por defecto; sample_size (proporción observaciones muestreadas), cuyo valor por defecto es 1.0; y stop_iter (número de iteraciones antes de parar), en el que el valor por defecto es infinito [45].

2.4.5. Ajuste de hiperparámetros

En función de los resultados obtenidos y con el objetivo de mejorar el performance del modelo, en ocasiones se realiza lo que se conoce como ajuste de hiperparámetros [46]. Un hiperparámetro es una propiedad como tal de un algoritmo de aprendizaje que usualmente es un valor numérico (aunque también puede ser nominal) y debe ser configurado puesto que no es aprendido por el algoritmo. Al modificar un hiperparámetro el algoritmo cambia la forma como opera. El proceso de ajuste de los hiperparámetros es experimental y lo que se busca es determinar la mejor combinación de valores para dicho hiperparámetro. Una de las técnicas utilizadas para realizar el ajuste de hiperparámetros es 'grid search', esta técnica lo que hace es generar tantos modelos como combinaciones de hiperparámetros exista, por ejemplo, si el modelo escogido tiene dos hiperparámetros, el primero es un vector de cinco valores numéricos y el segundo es un vector nominal de dos elementos, entonces se generarán 10 modelos que serán entrenados y en función de los resultados de las métricas escogidas se selecciona el mejor modelo. Adicionalmente, existen otras técnicas para este ajuste como random search y Bayesian optimization [47]. Para este trabajo se utilizó la técnica grid searh para el ajuste de los hiperparámetros del modelo RF, el mismo que tiene tres hiperparámetros principales: a) m_{try}, número de predictores en cada división, b) $trees$, número de árboles, y c) $\min\_n$, número mínimo de observaciones en un nodo para continuar con la división. De estos tres hiperparámetros solamente se consideraron los dos últimos y se utilizó el paquete tune [45].

2.5. Evaluación

Dado que este trabajo corresponde a un problema de clasificación de ML, se consideraron las métricas que corresponden a este tipo de problema. Previamente, se debe tener en claro los cuatro casos que arrojan los resultados [48]. Para ejemplificar de mejor manera estos cuatro casos se asume un problema de clasificación en el que se aplica ML para determinar si un e-mail es spam o no spam.

- Verdadero positivo (TP), la predicción indica que es spam y el e-mail efectivamente es spam.
- Verdadero negativo (TN), la predicción es no spam y el e-mail efectivamente es no spam.
- Falso positivo (FP), la predicción indica que es spam cuando realmente el e-mail no es spam.
- Falso negativo (FN), la predicción india que no es spam cuando realmente si es.

Las métricas tomadas en cuenta para este proyecto son:

a) Accuracy, que se define como la relación de las etiquetas correctamente predecidas por el modelo con respecto a todo el conjunto de etiquetas [49]. Esta métrica se calcula con la siguiente fórmula:

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)}$$

Accuracy responde a la pregunta, ¿Cuántos e-mails con spam fueron etiquetados correctamente de todos los e-mails con spam?

b) Sensitivity o Recall, se refiere a la proporción de todos los casos positivos que fueron correctamente clasificados [50] y se la calcula de la siguiente manera:

$$Sensitivity = \frac{TP}{(TP + FN)}$$

En este caso, esta métrica responde a la pregunta, de todos los e-mails que son spam, ¿cuántos de ellos se predijeron corrrectamente?

c) Specificity, corresponde a la proporción de todos los casos negativos que fueron correctamente clasificados [51], dicha métrica se obtiene de la siguiente manera:

$$Specificity = \frac{TN}{(TN + FP)}$$

La pregunta que responde esta métrica es: de todos los e-mails que no son spam, ¿cuántos de ellos fueron predecidos correctamente?

d) Precision, es la relación de todos los casos que fueron predichos correctamente como positivos respecto a todos los etiquetados como positivos [52] y se calcula mediante la fórmula:

$$Precision = \frac{TP}{(TP + FP)}$$

La pregunta a ser respondida por esta métrica es: ¿cuántos de los e-mails que fueron predichos como spam son en realidad spam?

e) F1-Score o F-Measure, considera tanto precision como recall y se la considera a esta métrica como el promedio armónico de precision y recall [53]. Para su interpretación, se debe tener en cuenta que, si existe cierto equilibrio en precision y recall, la puntuación de esta métrica es mejor; por otro lado, si una de estas dos métricas mejora a expensas de la otra entonces la puntuación F1-Score no será alta. La fórmula para calcularla es:

$$F1 - Score = 2\frac{Recall * Precision}{Recall + Precision}$$

f) ROC AUC (Area Under the ROC Curve), esta métrica se utiliza para visualizar el performance del modelo de clasificación a través de los umbrales de probabilidad y resulta de la combinación de la tasa de falsos positivos (1 - specificity) y la sensitivity, estas métricas son asignadas a los ejes X y Y respectivamente [54]. Cabe mencionar que esta métrica solamente puede ser usada en modelos de clasificación que también determinan la probabilidad de la predicción como en el caso de los modelos SVM, RF y XGBoost. Mientras más alto es el valor de esta métrica, mejor es el performance del modelo.

2.6. Despliegue

El último paso en este trabajo es el desarrollo de un dashboard. Uno de los inconvenientes en los proyectos de ML es que los modelos no llegan a la etapa de producción o en el mejor de los casos solamente se emite un informe técnico que no es de utilidad para la toma de decisiones. En este contexto, el uso de métodos de visualización de los resultados del modelo de ML ayuda a entenderlos de mejor manera el alcance del mismo, así como también permite a los usuarios finales a interactuar directamente con el modelo de ML [55]. Para este fin, en este trabajo se incluyó la implementación de un dashboard en el cual el usuario pueda ingresar los valores de las variables del modelo y predecir el nivel de corrosión.

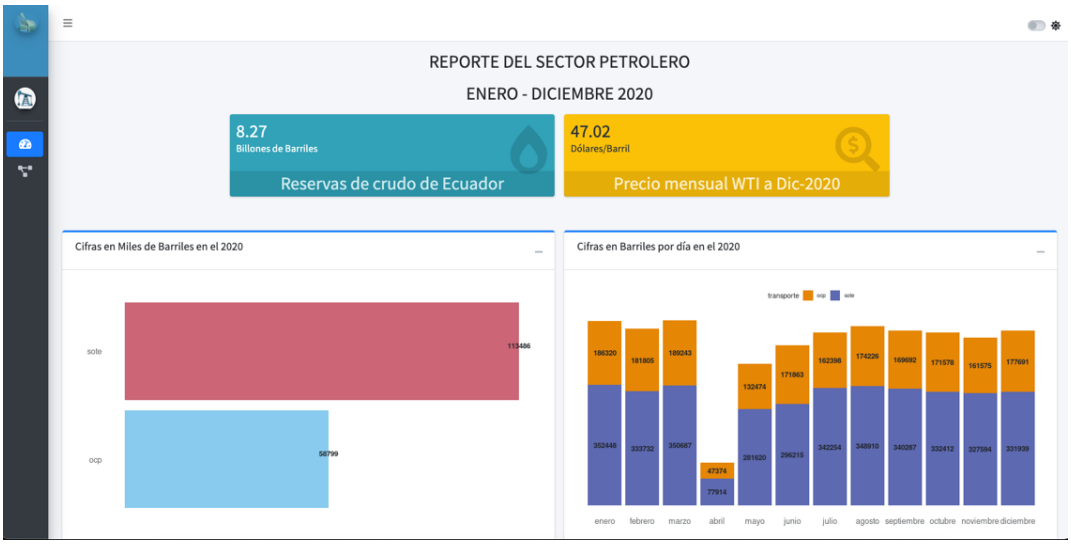

Figura 5 – Módulo de indicadores.

Este dashboard consta de dos partes, la primera corresponde a la presentación de indicadores del sector pretrolero del año 2020 (Figura 5), la segunda parte es en donde se ingresan los valores de las diferentes variables del modelo y luego al hacer clik en el botón Predicción, se presenta en un cuadro el resultado de dicha predicción que corresponde a uno de los cuatro niveles de criticidad definidos anteriormente (Figura 6).

El dashboard fue desarrollado en R mediante el uso de la librería Shiny. Este paquete es un marco referencial que permite crear aplicaciones web interactivas sin la necesidad de conocer HTML, CSS o JavaScript. Asimismo, la programación reactiva hace posible que, ante cualquier cambio en una entrada, la aplicación automáticamente descubra cómo actualizar las salidas correspondientes con la menor cantidad de trabajo [56]. Esta aplicación puede ser accedida desde cualquier explorador de internet a través del siguiente enlace: https://klever-mera.shinyapps.io/corrosion_pred/.

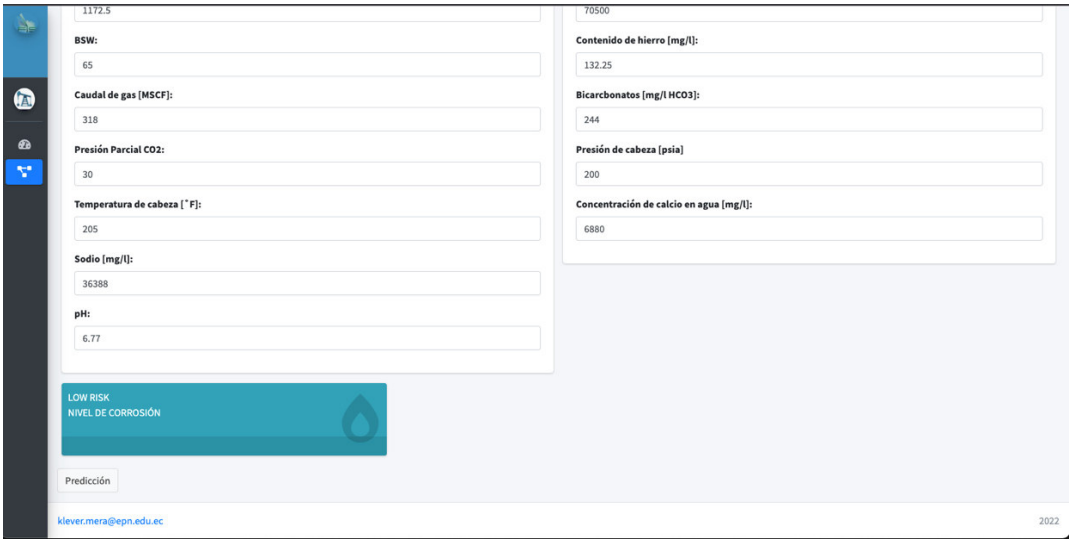

Figura 6 – Módulo de predicción.

3. RESULTADOS Y DISCUSIÓN

En esta sección se presentan los resultados de los modelos seleccionados para este trababjo (SVM, RF y XGBoost), con los que fue entrenado el conjunto de datos, el mismo que fue dividido 70% para entrenamiento y 30% para prueba. Asimismo, las métricas consideradas para este problema de clasificación también fueron previamente definidas y cuyos resultados se muestran en esta sección. El objetivo de los modelos de ML es predecir el nivel correspondiente de corrosión según los valores de las variables definidas como predictoras.

3.1. Resultados

Como se indicó previamente, en este trabajo se incluyó validación cruzada con k = 10, lo cual se evidencia en la Tabla 8, en la que se presenta las diez particiones y el tamaño de cada partición.

Tabla 8 – Validación cruzada con k = 10

Split	id
<split [3710/413]>	Fold01
<split [3710/413]>	Fold02
<split [3710/413]>	Fold03
<split [3711/412]>	Fold04
<split [3711/412]>	Fold05
<split [3711/412]>	Fold06
<split [3711/412]>	Fold07
<split [3711/412]>	Fold08
<split [3711/412]>	Fold09
<split [3711/412]>	Fold10

La Tabla 9 presenta los resultados obtenidos en el conjunto de datos de entrenamiento de los modelos SVM, RF y XGBoost, en dicha tabla se muestra la métrica, el estimador (promedio) y el modelo.

Tabla 9 – Métricas promedio de performance de los modelos SVM, RF y XGBoost

Métrica	Estimador	Modelo
Accuracy	0.751	SVM
F1-Score	0.716	SVM
Precision	0.665	SVM
Sensibility	0.844	SVM
Specificity	0.917	SVM

Métrica	Estimador	Modelo
Accuracy	0.951	RF
F1-Score	0.916	RF
Precision	0.899	RF
Sensitivity	0.938	RF
Specificity	0.979	RF
Accuracy	0.902	XGBoost
F1-Score	0.864	XGBoost
Precision	0.829	XGBoost
Sensitivity	0.914	XGBoost
Specificity	0.964	XGBoost

Para comparar el rendimiento de los tres modelos, la Tabla 10 presenta los valores mínimos, máximos y la mediana del estimador de cada métrica.

Tabla 10 – Métricas de performance de los modelos SVM, RF y XGBoost

Métrica	Mínimo	Mediana	Máximo	Modelo
Accuracy	0.697	0.763	0.775	SVM
F1-Score	0.649	0.714	0.768	SVM
Precision	0.598	0.666	0.741	SVM
Sensitivity	0.817	0.837	0.879	SVM
Specificity	0.898	0.917	0.929	SVM
Accuracy	0.932	0.952	0.966	RF
F1-Score	0.877	0.916	0.952	RF
Precision	0.855	0.886	0.955	RF
Sensitivity	0.903	0.942	0.955	RF
Specificity	0.968	0.980	0.989	RF
Accuracy	0.888	0.903	0.920	XGBoost
F1-Score	0.828	0.867	0.895	XGBoost
Precision	0.783	0.821	0.880	XGBoost
Sensitivity	0.872	0.920	0.938	XGBoost
Specificity	0.958	0.965	0.969	XGBoost

También se incluyó en este trabajo el ajuste de hiperparámetros del modelo RF, cuyos resultados se presentan en la Tabla 11.

Tabla 11 – Ajuste de hiperparámetros del modelo RF

trees	min_n
1933	6
898	39
521	34
903	33
528	19

Luego de que el modelo fue re-entrenado teniendo en cuenta el ajuste de hiperparámetros, los resultados de las métricas de rendimiento del modelo final se muestran en la Tabla 12. Como se puede apreciar, por cada combinación de los hiperparámetros se genera un resultado de cada métrica y dado que son cinco métricas y cinco combinaciones de los dos hiperparámetros, se obtienen 25 resultados.

Tabla 12 – Métricas de performance luego del ajuste de hiperparámetros

trees	min_n	Métrica	Estimador	Modelo
1933	6	Accuracy	0,951	Modelo1
1933	6	F1-Score	0,915	Modelo1
1933	6	Precision	0,898	Modelo1
1933	6	Sensitivity	0,936	Modelo1
1933	6	Specificity	0,978	Modelo1
898	39	Accuracy	0,943	Modelo2
898	39	F1-Score	0,905	Modelo2
898	39	Precision	0,883	Modelo2
898	39	Sensitivity	0,934	Modelo2
898	39	Specificity	0,976	Modelo2
521	34	Accuracy	0,945	Modelo3
521	34	F1-Score	0,909	Modelo3
521	34	Precision	0,888	Modelo3
521	34	Sensitivity	0,936	Modelo3
521	34	Specificity	0,977	Modelo3
903	33	Accuracy	0,947	Modelo4
903	33	F1-Score	0,911	Modelo4
903	33	Precision	0,890	Modelo4
903	33	Sensitivity	0,937	Modelo4
903	33	Specificity	0,978	Modelo4
528	19	Accuracy	0,951	Modelo5
528	19	F1-Score	0,915	Modelo5
528	19	Precision	0,896	Modelo5
528	19	Sensitivity	0,939	Modelo5
528	19	Specificity	0,979	Modelo5

A continuación, se presentan las gráficas del ajuste de hiperparámetros en las que se puede ver que son diez gráficas ya que son dos hiperparámetros, trees y min_n, y cinco métricas. Además, en el eje X se muestran los valores de ajuste y en el eje Y los valores correspondientes a cada métrica (Figura 7).

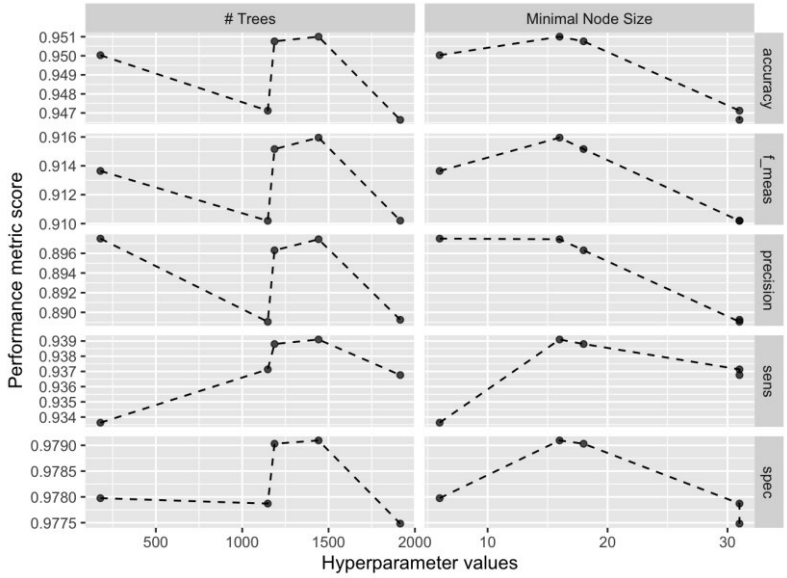

Figura 7 – Hiperparámetros y métricas.

Para seleccionar el mejor modelo, a continuación, se presenta los resultados de rendimiento de los cinco mejores modelos (Tabla 13). Para la generación de estos resultados se considera la métrica Sensitivity puesto que es la que se requiere mejorar.

Tabla 13 – Mejores cinco modelos luego del ajuste de hiperparámetros

trees	min_n	Métrica	Estimador	Modelo
528	19	Sensitivity	0.939	Modelo5
903	33	Sensitivity	0.937	Modelo4
1933	6	Sensitivity	0.936	Modelo1
521	34	Sensitivity	0.936	Modelo3
898	39	Sensitivity	0.934	Modelo2

Una vez seleccionado el mejor modelo, según los resultados del ajuste de hiperparámetros, el paso final es re-entrenar el conjunto de datos de entrenamiento con este nuevo modelo y obtener los resultados de la métrica ROC AUC. En la Figura 8 se muestra la curva ROC AUC por cada uno de los niveles de criticidad de la corrosión y en la Tabla 14 el valor

calculado, el mismo que fue determinado utilizado el método de Hand y Till puesto que este proyecto es una clasificación multiclase [57].

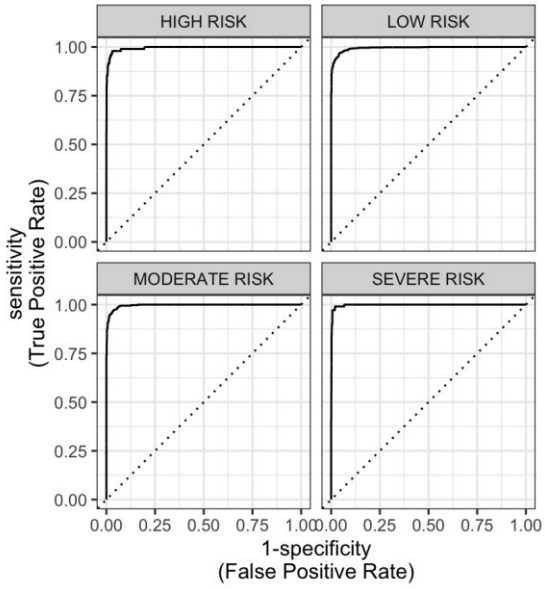

Figura 8 – ROC AUC.

Tabla 14 – ROC AUC para multi clase

Métrica	Estimador	Método
roc_auc	0.996	hand_till

Finalmente, la gráfica de la importancia de las características, la misma que muestra solamente 12 de las 14 variables predictivas, esto se debe a que en la etapa de ingeniería de características se configuró un paso para que cuando la correlación sea mayor o igual a 0.9 sea eliminada una de las variables, razón por la cual tanto el cloro (Cl) como el calcio (Ca) ya no aparecen en la Figura 9.

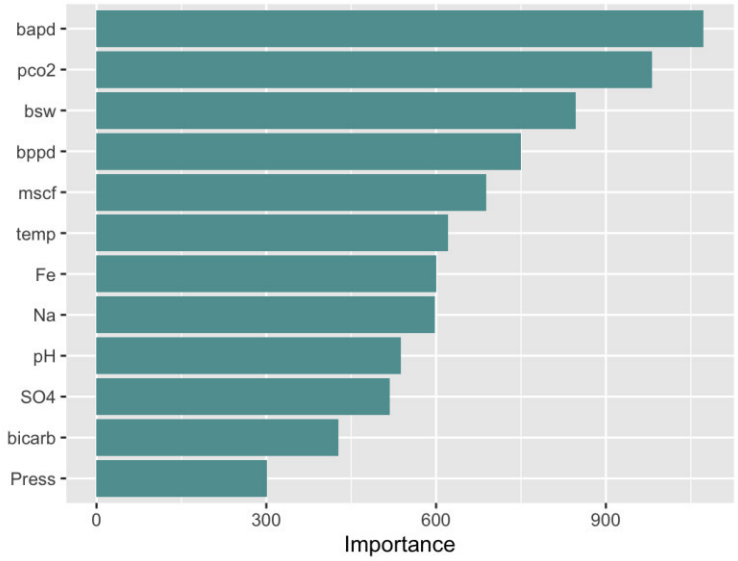

Figura 9 – Importancia de características.

3.2. Discusiones

El proceso de modelado partió de tres modelos, SVM, RF y XGBoost, luego del entrenamiento en el flujo de trabajo y considerando tanto el recipiente de ingeniería de características así como también la validación cruzada con k = 10, cada métrica es estimada diez veces y luego se obtiene el valor promedio, que es lo que se presenta en la tabla 9. De dicha tabla se aprecia que las métricas del modelo SVM son las más bajas y las del modelo RF las más altas. Luego, al considerar las métricas de la tabla 10, que es un resumen de los resultados de la validación cruzada, en la que se presentan los valores mínimos, máximos y mediana, se concluye que definitivamente los resultados del modelo RF son mejores respecto a los modelos SVM y XGBoost y por lo tanto se seleccionó a RF para el ajuste de hiperparámetros.

Los resultados en promedio de cada una de las métricas considerando las diferentes combinaciones de los dos parámetros seleccionados para su ajuste (tabla 12), indican que los modelos 1 y 5 presentan los mejores resultados. Asimismo, en la figura 7 se aprecia el efecto en las métricas de las diferentes combinaciones de los hiperparámetros, es así como, los cambios en el número de árboles (trees) alcanza el valor más alto para cada métrica cuando este hiperparámetro tiene un valor cercano a 1500, ya que luego de este valor el performance empieza a degradarse. Respecto al tamaño mínimo del nodo (min_n) inicia con un performance bajo (excepto para precision) y luego conforme se incrementa su valor, hasta llegar a seis, alcanza el valor más alto en todas las métricas para luego empezar a degradar dichas métricas.

A continuación, se debe seleccionar el mejor modelo en función del performance de cada uno de los candidatos, para lo cual, y con la ayuda de la función 'show_best', en la tabla 13 se muestran los cinco mejores modelos en base al valor promedio de la métrica, es así que el modelo 5 resultó ser el mejor con los hiperparámetros trees de 528 y min_n de 19. Este modelo final se incluyó en el flujo de trabajo, se entrenó nuevamente y se obtuvo la métrica final que para este caso se seleccionó la curva ROC AUC cuyos resultados se aprecian tanto en la figura 8 como en la tabla 14, de los cuales se desprende que se han conseguido buenos resultados ya que las curvas como tal están cerca del valor de 1 y el performance promedio final tiene un valor alto de 0.996.

Otra de las actividades que se llevó a cabo en este trabajo es el análisis de la importancia de las características, en la figura 9 se aprecia que la variable predictora que mayor importancia tiene para el modelo es el caudal de agua (bapd), seguida por la presión parcial CO2 (pco2), el sedimento básico y agua (bsw), el caudal de petróleo (bppd) y el caudal de gas (mscf). En función de estos resultados se puede indicar que todas las variables predictoras asociadas con la producción (tabla 1) se ubican en los primeros cinco lugares mientras que aquellas asociadas a la química solamente pco2 se encuentra en los primeros lugares. Además, los últimos tres lugares en importancia les corresponden a las variables químicas concentración de sulfatos en agua (SO4), bicarbonatos (bicarb) y presión (Press).

4. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

Pese a todos los inconvenientes de la disponibilidad de los datos (ubicación, estructura, disposición y periodicidad), que se encontraban distribuidos en muchos archivos de Excel, los procesos de extracción y transformación fueron lo suficientemente robustos puesto que permitieron generar un conjunto de datos consolidado que cumple con las condiciones de que cada variable una columna y cada observación una fila, de tal manera que los datos han sido dispuestos de manera organizada lo cual benefició a los procesos posteriores como el análisis, modelado y puesta en producción.

La metodología CRISP-DM se aplicó en este trabajo durante todo el ciclo de vida del proyecto de datos, la misma que gracias a sus seis fases permite conocer desde el inicio las necesidades del negocio, los datos que se requieren juntamente con su preparación y limpieza, luego el modelado la evaluación y el desarrollo con lo que se consiguió llevar a delante un proyecto de inicio a fin con las ventajas de ser una metodología de fácil implementación y flexible.

Un punto clave en este proyecto fue la ingeniería de características ya que en base al conjunto de datos que fue considerado para el modelado, se pudo crear un recipiente que en el que se incluyeron las tareas de normalización de todas las variables numéricas, la eliminación de variables cuya correlación supere 0.9 y finalmente el balanceo del conjunto de datos, que se consiguió con la aplicación de la técnica SMOTE, todas estas actividades, más la validación cruzada, aportaron al modelo no solo a alcanzar un alto performance sino también a que pueda generalizar de mejor manera cuando reciba nuevos datos.

Al ser la ciencia de datos parte de un entorno experimental es recomendable que justamente se realicen pruebas con diferentes modelos de tal manera de comparar los resultados obtenidos, según las métricas definidas, y así escoger aquel que mejores resultados entregue según el problema que se busca resolver. Para este trabajo se definieron tres modelos, SVM, RF y XGBoost, que efectivamente fueron puestos a prueba y función de su performance se seleccionó uno de ellos para las siguientes etapas de modelado, como el ajuste de hiperparámetros.

Respecto a las métricas de evaluación del modelo se escogieron seis (accuracy, f1-meas, precision, sensitivity, specificity y curva ROC AUC) cuyo enfoque está en los problemas de clasificación. Los resultados de dichas métricas posibilitaron seleccionar de manera sustentada el mejor modelo, así como también evidenciar de manera gráfica los resultados de dichas métricas en el conjunto de datos de prueba.

4.2. Recomendaciones

En este trabajo el número de variables no fue un problema para ninguno de los modelos seleccionados; sin embargo, en el caso de que se incremente la cantidad de variables se recomienda utilizar técnicas de reducción de la dimensionalidad como Análisis de Componente Principal o PCA ya que de esta manera se reduce el esfuerzo computacional del modelo cuando se realiza el entrenamiento. Este paso también puede ser incluido en la etapa de la ingeniería de características.

Los flujos de trabajo o workflows permiten automatizar las tareas del modelado de tal manera que no sea necesario crear una estructura independiente para cada modelo, es por esto que se recomienda definir el flujo de trabajo y luego incluir el o los modelos junto con las métricas de tal manera que se optimiza el código, así como también la revisión y el control de las actividades del modelado.

Se recomienda también que en el ajuste de hiperparámetros se considere el uso de las otras técnicas como random search y luego realizar la comparación del resultado de las métricas definidas para el problema de tal manera de medir el efecto en el performance del modelo.

Respecto al lenguaje de programación, todo el código fue generado en R, lo cual demuestra la potencia y bondades de dicho programa que cuenta con todas las librerías del caso, así como también los ecosistemas y marcos referenciales que permiten llevar a cabo un proyecto de datos de extremo a extremo como se lo hizo en este proyecto puesto que desde la extracción de datos hasta la aplicación web fueron totalmente desarrollados en R. Es por esto por lo que se recomienda considerar a R para el desarrollo de proyectos de datos.

REFERENCIAS BIBLIOGRÁFICAS

[1] G. Gómez Ponce, “Ingresos petroleros en Ecuador: ¿puede el país seguir sosteniendo su economía en el crudo?”, 2021. [Online]. Available: https://www.gastopublico.org/informes-del-observatorio/ingresos-petroleros-en-ecuador-puede-el-pais-seguir-sosteniendo-su-economia-en-el-crudo. [Accessed: 11- Oct-2021].

[2] Banco Central del Ecuador, “La economía ecuatoriana inicia la recuperación económica con una expansión del 2,8% en 2021”, 2021. [Online]. Available: https://www.bce.fin.ec/index.php/boletines-de-prensa-archivo/item/1431-la-economia-ecuatoriana-inicia-la-recuperacion-economica-con-una-expansion-del-2-8-en-2021. [Accessed: 11-Oct-2021].

[3] R. from Deepdyve, "Corrosion research round-up", *Anti-Corrosion Methods and Materials*, vol. 9, no. 9, pp. 246-249, 1962.

[4] K. Tamalmani and H. Husin, "Review on Corrosion Inhibitors for Oil and Gas Corrosion Issues", *Applied Sciences*, vol. 10, no. 10, p. 3389, 2020.

[5] A. Alamri, "Localized corrosion and mitigation approach of steel materials used in oil and gas pipelines – An overview", *Engineering Failure Analysis*, vol. 116, p. 104735, 2020.

[6] S. Kapusta, F. van den Berg, R. Daane and M. Place, "The Impact of Oil Field Chemicals on Refinery Corrosion Problems", 2021. [Online]. Available: https://onepetro.org/NACECORR/proceedings-abstract/CORR03/All-CORR03/NACE-03649/114494. [Accessed: 12-Oct- 2021].

[7] C. Ossai, "Advances in Asset Management Techniques: An Overview of Corrosion Mechanisms and Mitigation Strategies for Oil and Gas Pipelines", *ISRN Corrosion*, vol. 2012, pp. 1-10, 2012.

[8] P. Jiang, "Machine learning methods for corrosion and stress corrosion cracking risk analysis of engineered systems", Ph.D. dissertation, Science Dept., The University of New South Wales, Australia, 2018.

[9] S. Zukhrufany, “The utilization of supervised machine learning in predicting corrosion to support preventing pipelines leakage in oil and gas industry”, M.S. thesis, Sc. and Tech. Dept., University of Stavanger, Norway, 2018.

[10] C. Ossai, “A data-driven machine learning approach for corrosion risk assessment—A comparative study”, *Big Data and Cognitive Computing*, vol. 3, no. 2, p. 28, 2019.

[11] L. Coelho, D. Zhang, Y. Van Ingelgem, D. Steckelmacher, A. Nowé, & H. Terryn, “Reviewing machine learning of corrosion prediction in a data-oriented perspective”, *Materials Degradation*, vol. 6, p. 1, 2022.

[12] A. Hernández, “Fundamentos de Aseguramiento de Flujo en Sistemas de Producción de Petróleo y Gas”, disertación de ingeniería, Facultad de Ingeniería, Universidad Nacional Autónoma de México, México D.F., 2014.

[13] I. Acuña, A. Monsegue, T. Brill, H. Graven, F. Mulders, J. Le Calvez, E. Nichols, F. Bermudez, D. Notoadinegoro, & I. Sofronov, “Scanning for downhole corrosion”, *Oilfield Review*, vol. 22, pp.42-50, 2010.

[14] M. Abbas, “Modelling CO2 Corrosion of Pipeline Steels” ", Ph.D. dissertation, Facultity of Science, Agriculture and Engineering., Newcastle University, United Kingdom, 2016.

[15] I. El Naqa, & M. Murphy, “What Is Machine Learning?”, *Machine Learning in Radiation Oncology*, pp.3-11, 2015.

[16] M. Taddy, *Business Data Science.* New York: McGraw-Hill Education, 2019.

[17] A. Burkov, *The Hundred-Page Machine Learning Book*. Andriy Burkov, 2019.

[18] R. Bickham, “The Future of Machine Learning on Corrosion”, 2020. [Online]. Available: https://www.materialsperformance.com/articles/material-selection-design/2020/09/the-future-of-machine-learning-on-corrosion. [Accessed: 15-Ene-2022].
[19] S. Suthahran, *Machine Learning Models and Algorithms for Big Data Classification*. Springer Science+Business Media New York 2016.
[20] W. Noble, “What is a support vector machine?”, *Nature biotechnology*, vol. 24, no. 12, pp. 1565-1567, 2006.
[21] G. Biau, & E. Scornet, “A random forest guided tour”, *Test*, vol. 25, no. 2, pp. 197-227, 2016.
[22] A. Géron, *Hands-On Machine Learning with Scikit-Learn & TensorFlow: concepts, tools, and techniques to build intelligent systems*. Beijing: O`Reilly, 2019.
[23] S. Ramraj, N. Uzir, R. Sunil & S. Banerjee, “Experimenting XGBoost algorithm for prediction and classification of different datasets”, *International Journal of Control Theory and Applications*, vol. 9, p. 651-662, 2016.
[24] T. Chen, & C. Guestrin, “Xgboost: A scalable tree boosting system”, *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, p. 785-794, 2016.
[25] R. Wirth, & J. Hipp, “CRISP-DM: Towards a standard process model for data mining”, *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining,* vol. 1, London, UK: Springer-Verlag. 2000.
[26] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, & R. Wirth, "The CRISP-DM user guide", *4th CRISP-DM SIG Workshop in Brussels in March*, vol. 1999, pp. 1-14, 1999.
[27] R. Team, “R data import/export”, Version 3.2.3, 2015.
[28] H. Wickham, “Tidy Data”, *Journal of Statistical Software*, vol. 59, no. 10, pp. 1-23, 2014.
[29] A. Jadhav, D. Pramod & K. Ramanathan, “Comparison of performance of data imputation methods for numeric dataset”, *Applied Artificial Intelligence*, vol. 33, no. 10, pp. 913-933, 2019.
[30] G. Grolemund, & H. Wickham, “Dates and times made easy with lubridate”, *Journal of statistical software*, vol. 40, pp. 1-25, 2011.
[31] D. Bennett, “How can I deal with missing data in my study?”, *Australian and New Zealand journal of public health*, vol. 25, no. 5, pp. 464-469, 2001.
[32] H. Wickham, & M. Wickham, “Package ‘plyr’”, *Obtenido Httpscran Rproject Orgwebpackagesdplyrdplyr Pdf*, 2020.
[33] J. Emerson, W. Green, B. Schloerke, J. Crowley, D. Cook, H. Hofmann & H. Wickham, “The generalized pairs plot”, *Journal of Computational and Graphical Statistics*, vol. 22, no. 1, pp. 79-91. 2013.
[34] M. Kuhn & H. Wickham, “Tidymodels: a collection of packages for modeling and machine learning using tidyverse principles”, [Online]. Available: https://www.tidymodels.org. [Accessed: 17-Dic-2021].
[35] S. Dutta, A. Arunachalam & S. Misailovic, “To seed or not to seed? an empirical analysis of usage of seeds for testing in machine learning projects”, *IEEE Conference on Software Testing, Verification and Validation (ICST),* pp. 151-161, 2022.
[36] T. Topór, “Application of machine learning algorithms to predict permeability in tight sandstone formations”, *Naft. Gaz*, vol. 5, pp. 283-292, 2021.
[37] M. Kuhn & D. Vaughan, “Parsnip: A common api to modeling and analysis functions”, [R package version 0.1. 5], 2021.

[38] M. Uddin, J. Lee, S. Rizvi & S. Hamada, “Proposing enhanced feature engineering and a selection model for machine learning processes”, *Applied Sciences*, vol. 8, no. 4, p. 646, 2018.
[39] I. Guyon & A. Elisseeff, “An introduction to feature extraction”, *Feature extraction*, pp. 1-25, 2006.
[40] D. Borkin, A. Némethová, G. Michal’conok & K. Maiorov, “Impact of data normalization on classification model accuracy”, *Research Papers Faculty of Materials Science and Technology Slovak University of Technology*, vol. 27, no. 45, pp. 79-84, 2019.
[41] G. Batista, R. Prati & M. Monard, “A study of the behavior of several methods for balancing machine learning training data”, *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 20-29, 2004.
[42] A. Mohammed, M. Hassan, & D. Kadir, “Improving classification performance for a novel imbalanced medical dataset using SMOTE method”, *International Journal*, vol. 9, no. 3, pp. 3161-3172, 2020.
[43] J. Kim, & K. Chung, “Prediction model of user physical activity using data characteristics-based long short-term memory recurrent neural networks”, *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 13, no. 4, pp. 2060-2077, 2019.
[44] C. Ramezan, T. Warner & A. Maxwell, “Evaluation of sampling and cross-validation tuning strategies for regional-scale machine learning classification”, *Remote Sensing*, vol. 11, no. 2, p. 185, 2019.
[45] M. Kuhn & J. Silge, *Tidy Modeling with R, A Framework for Modeling in the Tidyverse*. O`Reilly, 2022.
[46] H. Weerts, A. Mueller & J. Vanschoren, “Importance of tuning hyperparameters of machine learning algorithms”, arXiv preprint arXiv:2007.07588, 2020.
[47] P. Probst, A. Boulesteix & B. Bischl, “Tunability: Importance of hyperparameters of machine learning algorithms”, *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 1934-1965, 2019.
[48] R. Choudhary & H. Gianey, “Comprehensive review on supervised machine learning algorithms”, *2017 International Conference on Machine Learning and Data Science (MLDS)*, pp. 37-43, 2017.
[49] M. Grandini, E. Bagli & G. Visani, “Metrics for multi-class classification: an overview”, *arXiv preprint arXiv:2008.05756*, 2020.
[50] T. Nguyen & G. Armitage, “A survey of techniques for internet traffic classification using machine learning”, *IEEE communications surveys & tutorials*, vol. 10, no. 4, pp. 56-76, 2008.
[51] H. Dalianis, “Evaluation metrics and evaluation”. *Clinical text mining,* pp. 45-53, Springer, Cham, 2018.
[52] A. Gunawardana & G. Shani, “A survey of accuracy evaluation metrics of recommendation tasks”, *Journal of Machine Learning Research*, vol. 10, no. 12, 2009.
[53] M. Hossin & M. Sulaiman, “A review on evaluation metrics for data classification evaluations”, *International journal of data mining & knowledge management process*, vol. 5, no. 2, p. 1, 2015.
[54] M. Tiwari, V. Sharma & D. Bala, “Credit Card Fraud Detection”, *JOURNAL OF ALGEBRAIC STATISTICS*, vol. 13, no. 2, pp. 1778-1789, 2022.
[55] K. Verbert, E. Duval, J. Klerkx, S. Govaerts, & J. Santos, “Learning analytics dashboard applications”, *American Behavioral Scientist*, vol. 57, no. 10, pp. 1500-1509, 2013.
[56] H. Wickham, *Mastering Shiny, Build Interactive Apps, Reports & Dashbords Powered by R*, O`Reilly, 2022.

[57] D. Hand & R. Till, “A simple generalisation of the area under the ROC curve for multiple class classification problems”, *Machine learning*, vol. 45, no. 2, pp. 171-186, 2001.

ANEXOS

Anexo I – Flujos de trabajo de cada modelo

```
══ SVM Workflow ═══════════════════════════════════════════════
Preprocessor: Recipe
Model: svm_rbf()
── Preprocessor ──────────────────────────────────────────────
3 Recipe Steps
• step_corr()
• step_normalize()
• step_smote()
── Model ─────────────────────────────────────────────────────
Radial Basis Function Support Vector Machine Specification (classification)
Computational engine: kernlab

══ RF Workflow ════════════════════════════════════════════════
Preprocessor: Recipe
Model: rand_forest()
── Preprocessor ──────────────────────────────────────────────
3 Recipe Steps
• step_corr()
• step_normalize()
• step_smote()
── Model ─────────────────────────────────────────────────────
Random Forest Model Specification (classification)
Engine-Specific Arguments:
  num.threads = parallel::detectCores()
Computational engine: ranger

══ XGBoost Workflow ═══════════════════════════════════════════
Preprocessor: Recipe
Model: boost_tree()
── Preprocessor ──────────────────────────────────────────────
3 Recipe Steps
• step_corr()
• step_normalize()
```

• step_smote()

── Model ──

Boosted Tree Model Specification (classification)

Engine-Specific Arguments:

  num.threads = parallel::detectCores()

Computational engine: xgboost

Anexo II – Información de la sesión en R

```
R version 4.1.3 (2022-03-10)
Platform: x86_64-apple-darwin17.0 (64-bit)
Running under: macOS Monterey 12.3.1

Matrix products: default
LAPACK: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRlapack.dylib

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base


other attached packages:
 [1] lubridate_1.8.0   stringr_1.4.0    readxl_1.4.0       tidyr_1.2.0        butcher_0.1.5
 [6] vip_0.3.2         dials_0.1.0      scales_1.1.1       xgboost_1.5.2.1    ranger_0.13.1
[11] kernlab_0.9-29    themis_0.1.4     workflows_0.2.6    ggplot2_3.3.5      yardstick_0.0.9
[16] tune_0.2.0        parsnip_0.2.1    recipes_0.2.0      dplyr_1.0.8        rsample_0.1.1
```