DISEÑO DE UN ALGORITMO DE DECISIÓN CON MACHINE LEARNING PARA LA OBTENCIÓN DE UNA RESPUESTA SOBRE LA APLICABILIDAD DE LOS EQUIPOS ESP, BASÁNDOSE EN EL ANÁLISIS ESTADÍSTICO DEL COMPORTAMIENTO DE ESTOS BAJO CONDICIONES ESPECIALES DE CAMPOS EN COLOMBIA

JUAN SEBASTIÁN ANDRADE RODRÍGUEZ

Proyecto integral de grado para optar al título de:

Ingeniero de Petróleos

Director:

Guillermo Triana Pérez

Ingeniero de Petróleos

FUNDACIÓN UNIVERSIDAD DE AMÉRICA
FACULTAD DE INGENIERÍAS
PROGRAMA DE INGENIERÍA DE PETRÓLEOS
BOGOTÁ D.C.
2021

NOTA DE ACEPTACIÓN

_____

_____

_____

_____

_____

_____

Nombre

Firma del Director

_____

Nombre

Firma del Presiente Jurado

_____

Nombre

Firma del Presiente Jurado

_____

Nombre

Firma del Presiente Jurado

_____

Nombre

Firma del Presiente Jurado

Bogotá D.C. Abril de 2021

DIRECTIVOS DE LA UNIVERSIDAD

Presidente de la Universidad y Rector de Claustro

Dr. Mario Posada García Peña

Consejero Constitucional

Dr. Luis Jaime Posada García-Peña

Vicerrectora Académica y de Investigaciones

Dra. Alexandra Mejía Guzmán

Secretaria General

Dr. José Luis Macías Rodríguez

Decano de la Facultad

Dr. Julio Cesar Fuentes Arismendi

Director de Programa

Dr. Juan Carlos Rodríguez Esparza

Las directivas de la Universidad de América, los jurados calificadores y el cuerpo docente no son responsables por los criterios e ideas expuestas en el presente documento. Estos corresponden únicamente a los autores.

TABLA DE CONTENIDO

pág.

LISTA DE FIGURAS

LISTA DE TABLAS

RESUMEN

La producción de petróleo mediante el uso de sistemas de bombeo electrosumergible es un proceso industrial que genera la necesidad recolectar y analizar una gran cantidad de información, que se almacena con el objetivo de ser utilizada como base estadística para futuros procesos. En este proyecto se realizó un estudio estadístico con *Machine Learning*, mediante la programación de un algoritmo de aproximadamente 2500 líneas de código, y utilizando la metodología *Caret* en el software RStudio. Se evaluaron un total de 586 pozos y 51 variables, de los cuales el 80% se utilizó para entrenar el modelo predictivo, y el 20% restante se utilizó para determinar la capacidad predictiva del modelo. Los resultados con el 80% permitieron observar la forma en la que está distribuida la probabilidad en los niveles de la variable "Causa_Raíz". Con los resultados del 20% se programaron gráficos que permitieron observar los resultados de la predicción en función de las condiciones especiales de campo arena, asfalteno y scale. Lo anterior permitió establecer que el algoritmo tiene la capacidad de predecir valores diferentes a los originales. El estudio se fortalece con el criterio y el conocimiento del ingeniero de petróleos, ya que permite analizar el resultado final del estudio, de tal forma que se logre tomar la mejor decisión sobre la aplicabilidad de los equipos.

Palabras clave: Bombeo Electrosumergible, Condiciones Especiales Campo, Producción, Match, Teardown, Causa Raíz, *Machine Learning*.

INTRODUCCIÓN

El *Machine Learning* es la rama de la *Inteligencia Artificial (IA)* que ha logrado revolucionar el mundo de los datos, a partir de la unión entre el campo de las ciencias informáticas y el campo de la estadística. Esta metodología suele implementarse mediante la construcción de algoritmos basados en la experiencia acumulada, con la finalidad de mejorar automáticamente la eficiencia de la solución de problemas asociados al análisis de *Big Data*, como el reconocimiento de patrones, las regresiones y el *clustering*. Esto ha llevado a que múltiples industrias puedan hacer un uso inteligente de la información que almacenan de cada uno de los procesos industriales a los que se dedican.

Uno de los procesos industriales llevados a cabo en la industria de los hidrocarburos es la producción de crudo mediante equipos de Bombeo Electrosumergible (ESP), un tipo de Sistema de Levantamiento Artificial (SLA) que requiere de actividades conjuntas para que su implementación sea óptima, lo que lleva a las compañías a almacenar grandes cantidades de información relacionadas a las propiedades del yacimiento, características del pozo, datos de producción, parámetro de diseño, y comportamiento de los equipos ESP durante las operaciones. A raíz de la complejidad de estas actividades, la información a almacenar aumenta cuando los equipos ESP presentan fallas durante las operaciones. En este proyecto las fallas se clasifican según su causa raíz; cuando es una *falla directa* el equipo ESP presenta fallas mecánicas o eléctricas en uno o varios de sus componentes, y cuando es una *falla indirecta* el equipo presenta problemas de eficiencia y daños físicos en los componentes debido a altas temperaturas, agentes corrosivos, incrustaciones o taponamientos por arenas, crudos pesados que precipitan orgánicos y crudos livianos con un alto GOR.

Estudios demuestran que la técnica de *Machine Learning* analiza la información de tal forma que permite reconocer patrones de fallas en los datos almacenados de las operaciones de producción de crudo con equipos ESP. Esto lleva a las compañías dedicadas a la fabricación de este tipo de SLA a utilizar métodos estadísticos (como el *Machine Learning)*, para analizar la información que acumulan sobre el comportamiento de los equipos durante las operaciones, con el fin de predecir comportamientos futuros y así mejorar la eficiencia, tanto de la selección como del diseño de los equipos ESP para un pozo en específico.

El objetivo general de este proyecto consiste en diseñar un algoritmo de decisión con *Machine Learning* para la obtención de una respuesta sobre la aplicabilidad de los equipos ESP, basándose en el análisis estadístico del comportamiento de estos bajo condiciones especiales de campos en Colombia. El estudio se realiza con la información de la compañía prestadora de servicios *Compañía A* que opera en Colombia, de la cual se seleccionan un total de 586 pozos ubicados en campos operados por diferentes compañías, razón por la cual la información se trabaja confidencialmente. La metodología que se desarrolla a continuación con el fin de dar cumplimiento a los objetivos del proyecto está basada en una de las metodologías de *Machine Learning* que comúnmente se desarrollan con el software RStudio, un software libre de programación dedicado a la computación estadística. El primer objetivo específico consiste en construir un *dataframe* (conjunto de datos estructurados) que incluya datos de cada una de las variables categorizadas en los siguientes grupos: producción, condiciones especiales de campo, componentes del equipo ESP, *Match*, y *Teardown*. El segundo objetivo específico abarca un estudio estadístico que consiste en realizar un análisis exploratorio de los datos, haciendo uso de la estadística descriptiva para indagar a fondo en cada una de las variables y así poder determinar cuáles son las variables cualitativas y sus niveles, cual es la distribución de las variables cuantitativas, y cuál es la "variable respuesta" que se va a predecir; información indispensable para el algoritmo de *Machine Learning*. También abarca el preprocesado de los datos, selección de predictores y programación de varios modelos de *Machine Learning*.

En el desarrollo del tercer objetivo específico se realiza el entrenamiento del modelo predictivo y la ejecución de la predicción con el conjunto de prueba. Para lograr cada uno de estos pasos, previamente se debe separar el *dataframe* en dos grupos seleccionados aleatoriamente: 80% para los datos de entrenamiento del modelo y 20% para los datos de prueba del modelo (*Test*). Los resultados de este objetivo se representan gráficamente mostrando y relacionando la información de las variables Bomba, Motor y Tipo de Motor, con los resultados de las predicciones asociados a las condiciones especiales de campo. El cuarto objetivo específico consiste en validar el objetivo general de este proyecto, en donde se analizan pozos nuevos (o conocidos pero seleccionados aleatoriamente) con la finalidad de dar una respuesta sobre si se recomienda o no, la instalación de equipos ESP de la Compañía A.

1. MARCO TEÓRICO

Este proyecto abarca terminologías en las áreas de bombeo electrosumergible, parámetros petrofísicos de yacimiento y programación estadística, los cuales se explican concisamente con el fin de comprender los conceptos aplicados para dar solución a la hipótesis de esta investigación.

1.1. Bombeo Electrosumergible

Los sistemas de bombeo electrosumergible, comúnmente llamados ESP, se consideran el sistema de levantamiento artificial (SLA) más eficiente para manejar grandes volúmenes de producción de líquidos provenientes del yacimiento. Su funcionamiento se basa en "emplear la energía eléctrica convertida en energía mecánica, con el fin de levantar una columna de fluido desde un nivel determinado hasta la superficie, mediante la acción rotacional de una bomba centrífuga de múltiples etapas, que es impulsada por un motor electrosumergible" [1]. Generalmente se implementan en pozos con alto índice de productividad, alta relación agua petróleo, baja relación gas petróleo, y tiene la capacidad de trabajar sobre un amplio rango de profundidades y caudales.

1.1.1. Componentes del equipo ESP

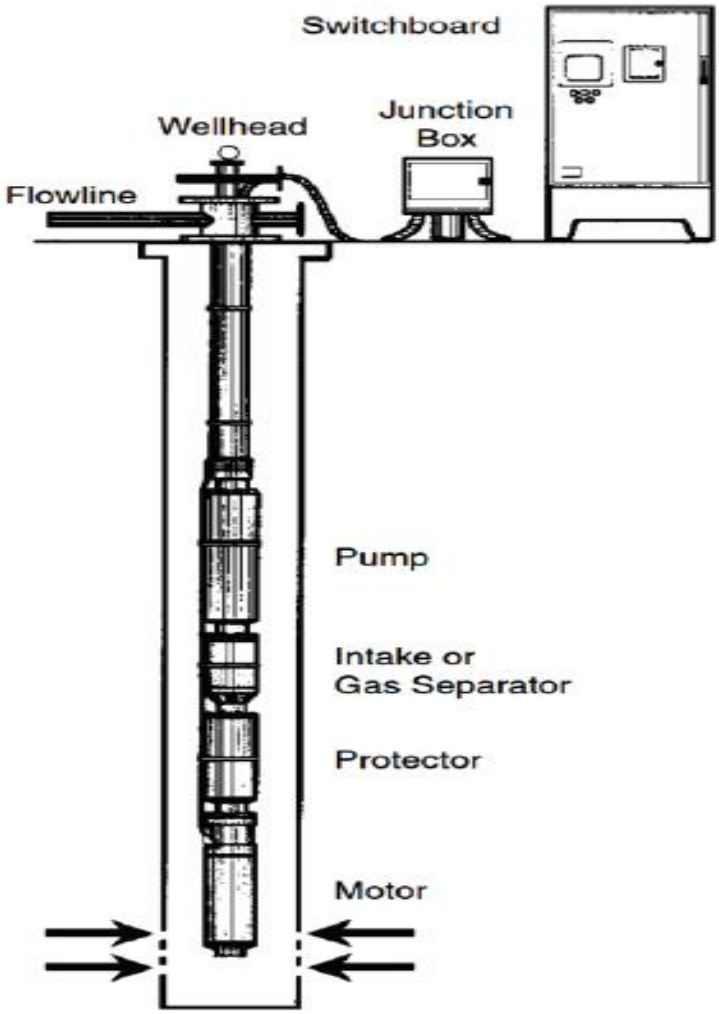
Los sistemas ESP están constituidos por un conjunto de equipos de superficie y un conjunto de equipos de fondo sumergidos dentro del pozo, como se observa en la **Figura 1.** En superficie se conecta un trasformador primario a la línea de alta tensión para reducir el voltaje a un nivel que pueda soportar un segundo equipo, el variador de frecuencia, el cual cumple las funciones de un transformador secundario que controla la frecuencia a la cual trabaja el motor. Las conexiones se hacen por medio de un cable aislado y resistente hasta que llega a la caja de conexiones, donde se conecta con el cable de potencia, el cual suministra la potencia eléctrica al motor en fondo. El cabezal de Pozo cumple la función de soportar el peso de los equipos de fondo y de mantener la presión anular del pozo en superficie [1].

El conjunto de equipos de fondo está formado por el cable de potencia, la bomba centrífuga multietapa, el separador de gas, el intake, el protector o sello, el motor y el sensor de fondo. Las funciones que cumple cada equipo mencionado anteriormente se resaltan en la **Tabla 1.** El sistema ESP también consta de otros equipos adicionales como: centralizador, válvula de

retención, válvula de purga, Y-Tool o Bypass y protectores para cable [2], los cuales se instalan según los requerimientos del pozo.

Figura 1.

Equipos del Sistema ESP



Nota. Los equipos en superficie se resumen en Variador de Frecuencia (Switchboard), la Caja de Conexiones (Junction Box) y el Cabezal de Pozo. Tomado de: F. Cachumba, *Estudio para la optimización de producción de pozos con bombeo electrosumergible, mediante análisis nodal del campo Cuyabeno*, Tesis Pregrado, Facultad de Ingeniería en Geología y Petróleos, Escuela Politécnica Nacional, Quito, Ecuador, 2017, [En linea]. Disponible: http://bibdigital.epn.edu.ec/handle/15000/18852.

Tabla 1.

Conjunto de Equipos de Fondo

Bomba Centrífuga	Bomba centrífuga formada por múltiples etapas, cada una con un impulsor rotatorio y un difusor estacionario, que provee la energía adicional para levantar el fluido mediante la generación de fuerzas centrífugas. El número de etapas determina la carga total generada, la potencia requerida y la altura de columna (TDH) deseada [1].
Intake	"Es una sección de más o menos 30 a 40 cm de longitud que genera la succión de fluidos hacia la bomba" [3]. Si se tiene capa de gas por debajo del nivel de profundidad del intake se instala un separador de gas para evitar que la disminuya la eficiencia del equipo.
Protector	Se encuentra ubicado entre el motor y el intake y cumple la función de aislar el motor de los fluidos de pozo, equilibrar la presión entre el aceite del motor y los fluidos del pozo, y transmitir el torque del motor a la bomba. Pueden ser de cámaras laberínticas o bolsas de elastómero [1].
Motor	"Provee la energía necesaria para que la bomba rote y acelere los fluidos que están siendo bombeados hacia la superficie" [2]. Pueden ser motores asincrónicos (AM) o motores de imanes permanentes (PMM).
Sensor	"Es un dispositivo electrónico que se encarga de enviar señales de presión y/o temperatura por medio del cable de potencia" [1].

Nota. Descripción de las funciones de cada uno de los componentes del equipo ESP que en conjunto logran levantar fluidos hasta superficie.

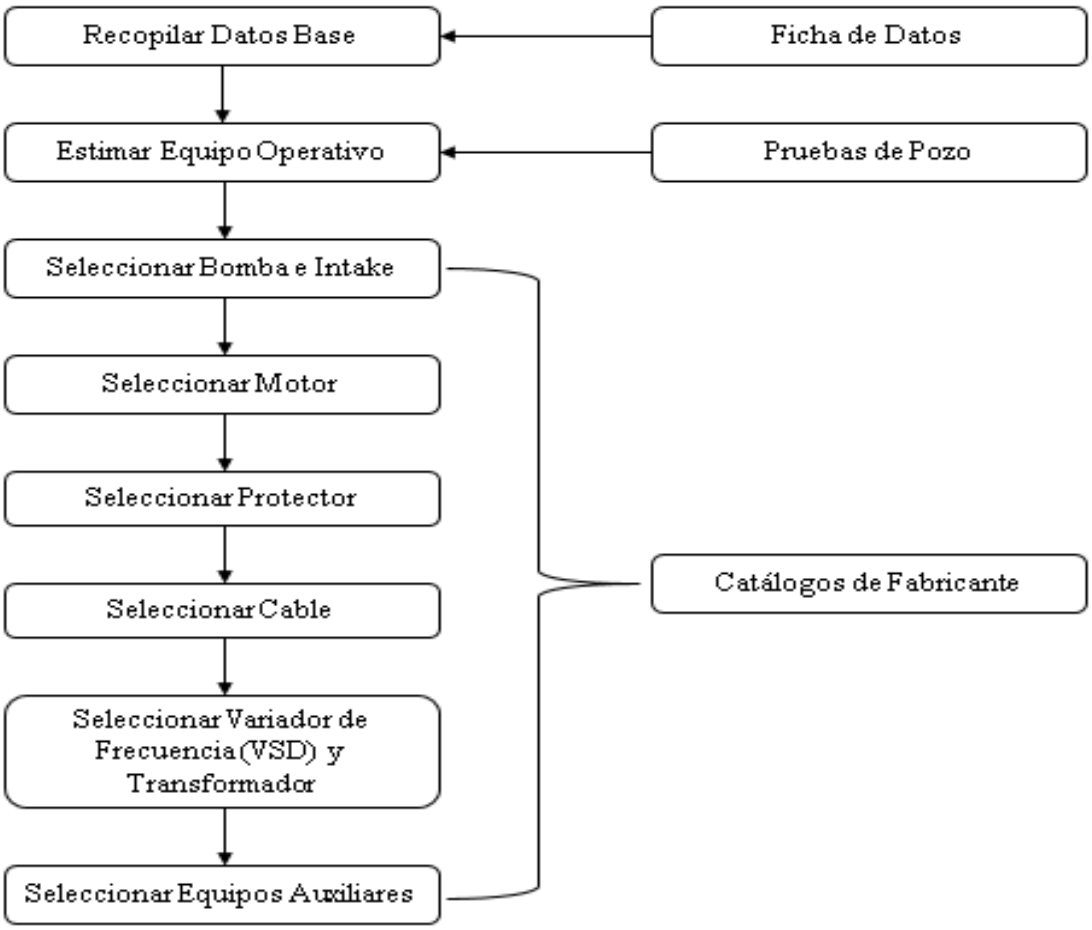
1.1.2. Diseño y Selección de Equipos ESP

Las compañías dedicadas a la fabricación de sistemas ESP deben basarse en la norma *API 11S4 Recommended Practice for Sizing and Selection of Electric Submersible Pum Installations* [4], la cual abarca desde los diseños básicos de aplicación del sistema, hasta los diseños de cada uno de sus componentes, tanto de superficie como de fondo. En este proyecto es importante explicar las consideraciones básicas de diseño y selección para tener claridad de los parámetros involucrados, ya que estos son los que se utilizan para la construcción del *dataframe* [4]**.** Estos se plasman en la *hoja de datos de diseño de ESP* adjunta en el **ANEXO 1**

1.1.2.i. Aplicación de Diseños Básicos. «El diagrama de flujo que se muestra en la Figura ***2.***, es una descripción general del procedimiento de diseño general de un equipo ESP. El diagrama ilustra un proceso lineal, pero en realidad se requiere de algunas iteraciones ya que un componente en particular puede impactar en un componente previamente seleccionado. También es posible que se desee realizar varias corridas de diseño para optimizar la selección del equipo respecto a los parámetros de entrada (input)» [4].

Figura 2.

Proceso de Diseño de Equipos ESP



Nota. Procedimiento de aplicación de los diseños básicos en el cual deben basarse los fabricantes de equipos ESP. Tomado de: Recommended Practice for Sizing and Selection of Electric Submersible Pump Installations, API 11S4, 3ª ed., American Petroleum Institute, Washington D. C., EE. UU., 2002. Modificado por J.S. Andrade.

1.1.2.ii. Recopilación de Datos Base. «Los datos se utilizan para describir el entorno en el que debe operar el equipo ESP, y su calidad se debe verificar de todas las fuentes disponibles, ya que de estos depende la calidad del diseño. Los datos relacionados al rendimiento del sistema de bombeo, el flujo de fluidos y las propiedades PVT del fluido se entienden fácilmente al modelarlos matemáticamente. Sin embargo, si se introducen datos inexactos en estos modelos, el valor de salida (*output*) será incorrecto, lo que conlleva a una selección y diseño de componentes inadecuados, y en última instancia conducen a una falla prematura del equipo» [4].

1.2. Condiciones Especiales de Campo

En la construcción del *dataframe* se agregan un conjunto de variables pertenecientes al grupo "Condiciones Especiales de Campo", el cual agrupa variables de yacimiento y variables propias de los fluidos, que se explican en la ***Tabla 2.***, y **Tabla *3.***, respectivamente. La explicación se hace en base a las afectaciones que estas variables causan sobre los equipos ESP durante las operaciones.

Tabla 2.

Variables de Yacimiento

Propiedad [Unidades]	Definición	Afectación a los Equipos ESP
Reservorio	"Se entiende por yacimiento una unidad geológica de volumen limitado, poroso y permeable que contiene hidrocarburos en estado líquido y/o gaseoso" [5].	Son las propiedades petrofísicas de cada reservorio las que causan afectaciones.
Temperatura de Yacimiento (Ty) [°F]	Es la temperatura a la cual se encuentran almacenados los fluidos en el yacimiento.	Si se tiene una Ty mayor a la estimada en el diseño del equipo, se puede generar una afectación física al material de cada componente.
Índice de Productividad (IP) [BFPD/psi]	Es una forma matemática de expresar la capacidad de un yacimiento para suministrar fluidos al pozo [6].	Amplios rangos en los valores de IP estimados por las operadoras general incertidumbre durante el diseño de equipos
Arenas [%] [ppm]	Fragmentos de roca sedimentaria tipo Arenisca que son arrastrados por los fluidos hacía el pozo, debido a formaciones poco consolidadas	Causan taponamiento en el intake y en la bomba. El contacto de la arena con el metal genera erosión que desgasta o rompe a los equipos.
Scale [%]	Se trata de un conjunto de depósitos de minerales que se incrustan a lo largo del camino que recorre el fluido. Esto sucede debido a la capacidad que tiene el agua de formación de disolver y transportar grandes cantidades de minerales [7].	Dichos minerales se incrustan en los poros de la formación, en los orificios de cañoneo, en el casing, en el intake, en la bomba, en el tubing, afectando la integridad física de estos y reduciendo parcial o totalmente el paso de fluidos.

Nota. Variables de yacimiento que hacen parte del grupo "Condiciones Especiales de Campo" que está contenido en el dataframe.

Tabla 3.

Variables del Fluido

Propiedad [Unidades]	Definición	Afectación a los Equipos ESP
Gravedad API [°API]	Escala de gravedad específica desarrollada por el American Petroleum Institute (API) para medir la densidad relativa de diversos líquidos de petróleo [1]. Clasifica los hidrocarburos según el ° API en extrapesados, pesados, medianos, livianos y condensados.	Crudos extrapesados y pesados tienen cadenas extensas de hidrocarburos, con componentes de gran densidad que precipitan y taponan los equipos de fondo. Crudos livianos que forman capa de gas a nivel de yacimiento disminuyen la eficiencia de la bomba y la taponan.
Viscosidad [cP]	"La viscosidad de un fluido es una medida de la fricción interna o resistencia que ofrecen sus moléculas a fluir" [8].	Un crudo de alta viscosidad provoca taponamiento del intake y de la bomba, fracturas en los componentes y un bajo levantamiento que sobrecarga el motor.
Relación Gas Petróleo (GOR) [PCN/BN]	Relación entre el volumen de gas producido y el volumen de petróleo producido.	No genera afectación siempre y cuando a la profundidad del intake el gas se encuentre disuelto en el petróleo.
Parafinas [%] [ppm]	Son compuestos orgánicos, conocidos como ceras parafínicas de largas cadenas de carbono, con una estructura molecular de macro cristales en forma de agujas [9].	Si se conglomeran constituyen grandes depósitos de cera que taponan el intake y la bomba, y aumentan la viscosidad del fluido generando una disminución en la eficiencia.
Asfaltenos [%] [ppm]	Son compuestos orgánicos de alto peso molecular (entre 1000 y 50000 uma), largas cadenas de carbonos y se encuentra suspendidos coloidalmente en el crudo por una capa estabilizante de resinas llamada micelas [9].	Fuerzas mecánicas o agentes químicos pueden desestabilizar la estructura micelar, ocasionado que los asfaltenos precipiten y taponen el intake y la bomba, o se adhieran a las superficies metálicas de los equipos.
Corrosión [%]	Presencia de azufre en la formación permite la formación de H_2S al reaccionar con el crudo. Si hay presencia de oxígeno este reacciona con el crudo formando CO_2.	El H_2S es un agente altamente corrosivo que afecta la integridad física de los equipos, disminuye la eficiencia y en ocasiones genera que el componente afectado se rompa.

Nota. Variables propias de los fluidos de yacimiento que hacen parte del grupo "Condiciones Especiales de Campo" que está contenido en el dataframe.

1.3. Machine Learning

El *Machine Learning*, que en español significa Aprendizaje Automático, consiste en una disciplina de las ciencias informáticas, relacionada con el desarrollo de la Inteligencia Artificial (IA), que hace referencia a la capacidad que tiene una máquina o un software para aprender a resolver automáticamente una serie de operaciones a partir de unos datos de entrada [10]. Su aplicación más importante en el mundo industrial es el manejo inteligente de la información, ya que permite analizar grandes cantidades de datos (*Big Data*) a un nivel tal que puede reconocer fácilmente patrones, regresiones y estructuras de datos o *Clustering* [11], con el fin de realizar predicciones futuras. En [12] Espinoza menciona que "la estadística es sin duda la base fundamental del aprendizaje automático, que básicamente consiste en una serie de algoritmos capaces de analizar grandes cantidades de datos para deducir cual es el resultado óptimo para un determinado problema". Por eso, es importante explicar en este proyecto tanto los conceptos básicos de la estadística, como el software a utilizar para la escritura y ejecución del algoritmo de *Machine Learning*.

1.3.1. Estadística

"La estadística es la ciencia que comprende una serie de métodos y procedimientos destinados a la recopilación, tabulación, procesamiento, análisis e interpretación de datos cuantitativos y cualitativos" [13]. Se divide en dos ramas, la estadística descriptiva y la estadística inferencial, y juntas tienen el objetivo de estudiar el comportamiento de una población o de un conjunto de datos.

1.3.1.i. Estadística Descriptiva. “Rama de la ciencia estadística que se encarga de la recopilación, procesamiento y análisis de la información siendo sus conclusiones válidas solo para el grupo analizado” [13].

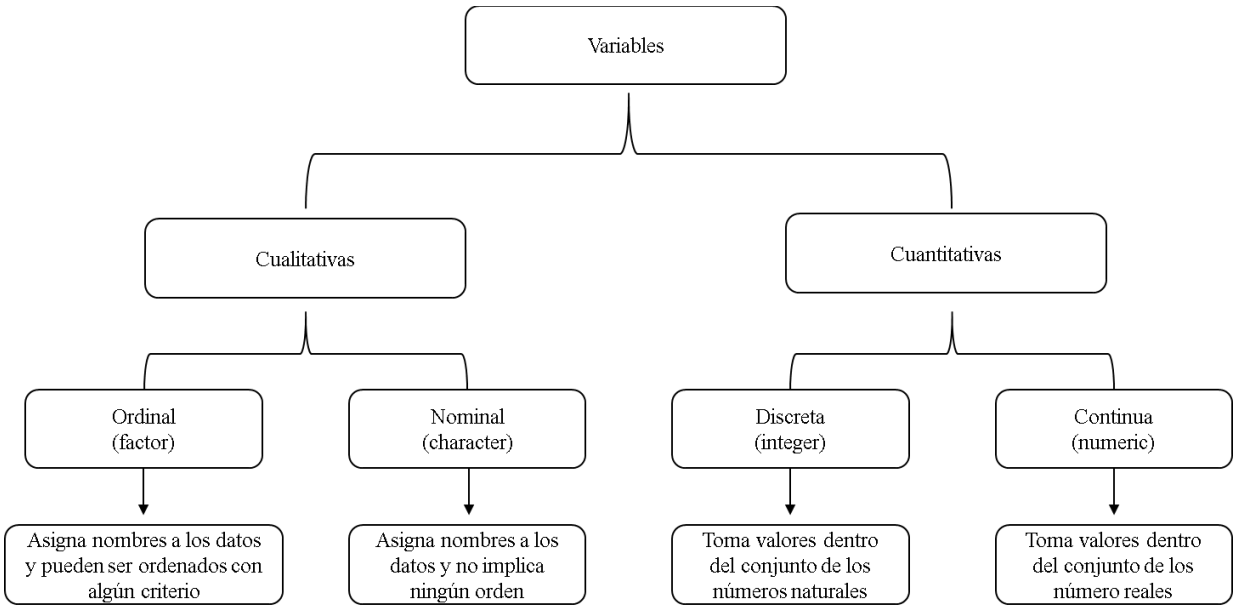
1.3.1.ii. Estadística Inferencial. “Rama de la ciencia estadística que proporciona métodos y procedimientos que permiten obtener conclusiones para una población a partir del estudio de una o más muestras representativas” [13].

1.3.1.iii. Dato. “Conocido también como información, es el valor de la variable asociada a un elemento de una población o una muestra” [13]. Un dato es cuantitativo cuando su valor es un valor numérico, es decir que es medible y que se puede contar. Un dato es cualitativo cuando su valor representa alguna característica o cualidad de los elementos de la población, es decir se representa como un dato tipo carácter o factor.

1.3.1.iv. Variable. Es una característica de la población o de la muestra cuya medida puede cambiar de valor [13]. En la **Figura 3.**, se observa cómo puede ser una variable según su naturaleza. Si los datos cualitativos de una muestra varían, se convierte en una “Variable Cualitativa”. De igual forma, si los datos cuantitativos de una muestra varían, se convierte en una “Variable Cuantitativa”.

Figura 3.

Clasificación de las Variables



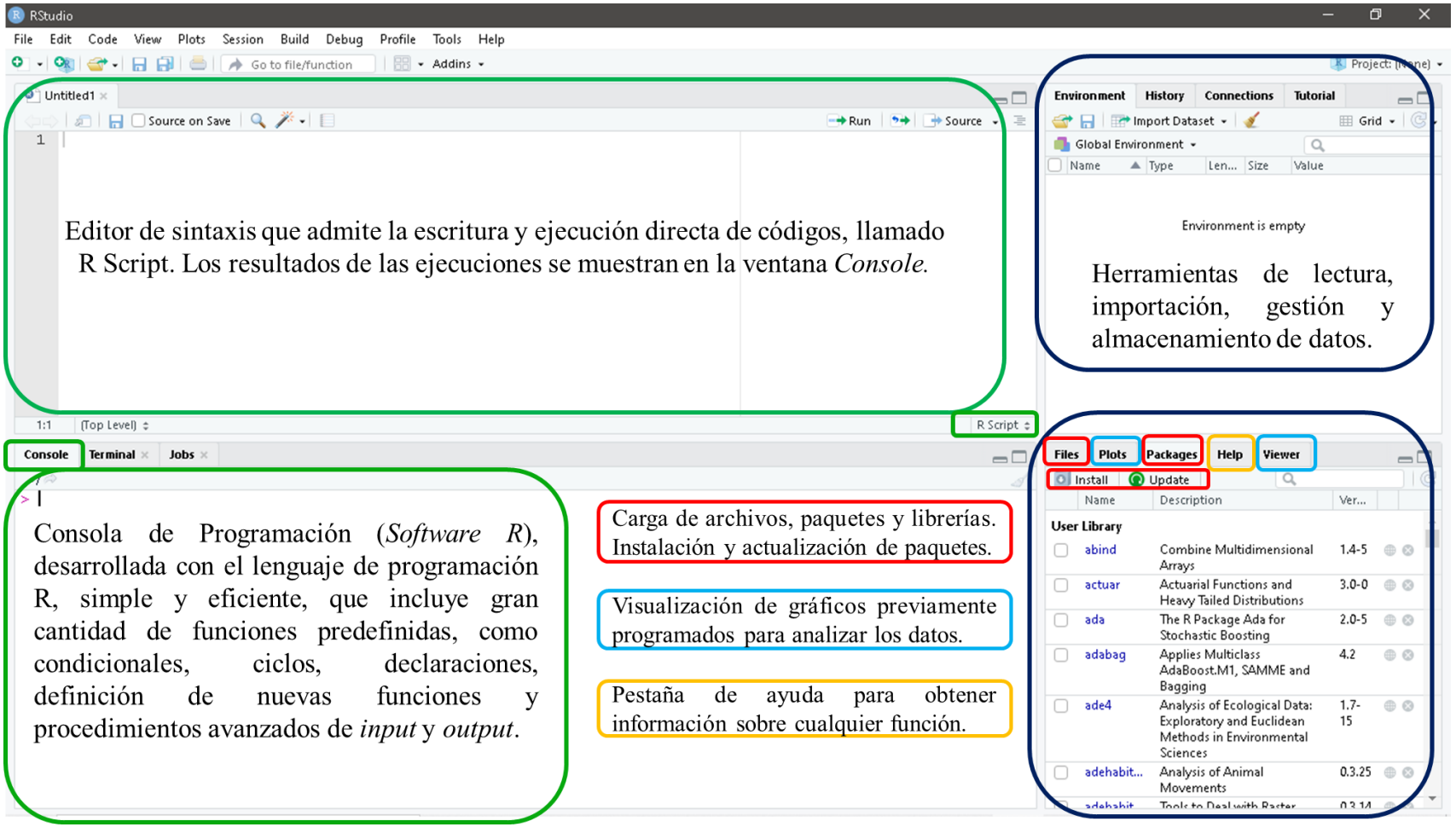
Nota. En el tercer nivel de clasificación se observan las divisiones de las variables cualitativas y cuantitativas. Cada una de estas tiene un término entre paréntesis (factor, character, integer y numeric) que hace referencia a la forma en que el programa RStudio almacena los tipos de datos.

1.3.2. Software RStudio

El software que contiene la consola de programación se llama *R*, un software libre y de código abierto utilizado para la computación estadística, el cual está formado por un conjunto de funciones que pueden aplicarse mediante la instalación de paquetes y librerías, o que pueden ser creadas directamente por el usuario [14]. Por otra parte, *RStudio* es un entorno de desarrollo integrado (IDE) diseñado para *R*, que incluye una consola de programación (software R), un editor de sintaxis que admite la ejecución directa de código (R Script), y un conjunto de herramientas [15] como se muestra en la **Figura 4.** Existen distintos paquetes de R para aplicar el *Machine Learning*, de los cuales se selecciona uno para el desarrollo de este proyecto. Los paquetes, las funciones y la metodología a seguir se explican en el capítulo de metodología y datos.

Figura 4.

Vista Preliminar de RStudio



Nota. Ventana preliminar que se abre al ejecutar el programa RStudio, explicada sección a sección. Explicaciones tomadas de: A. Santana. (2012). *Introducción al Entorno Estadístico*. [En línea]. Disponible en: https ://docplayer.es/70903696-Introduccion-al-entorno-estadistico-angelo-santana.html.

2. METODOLOGÍA Y DATOS

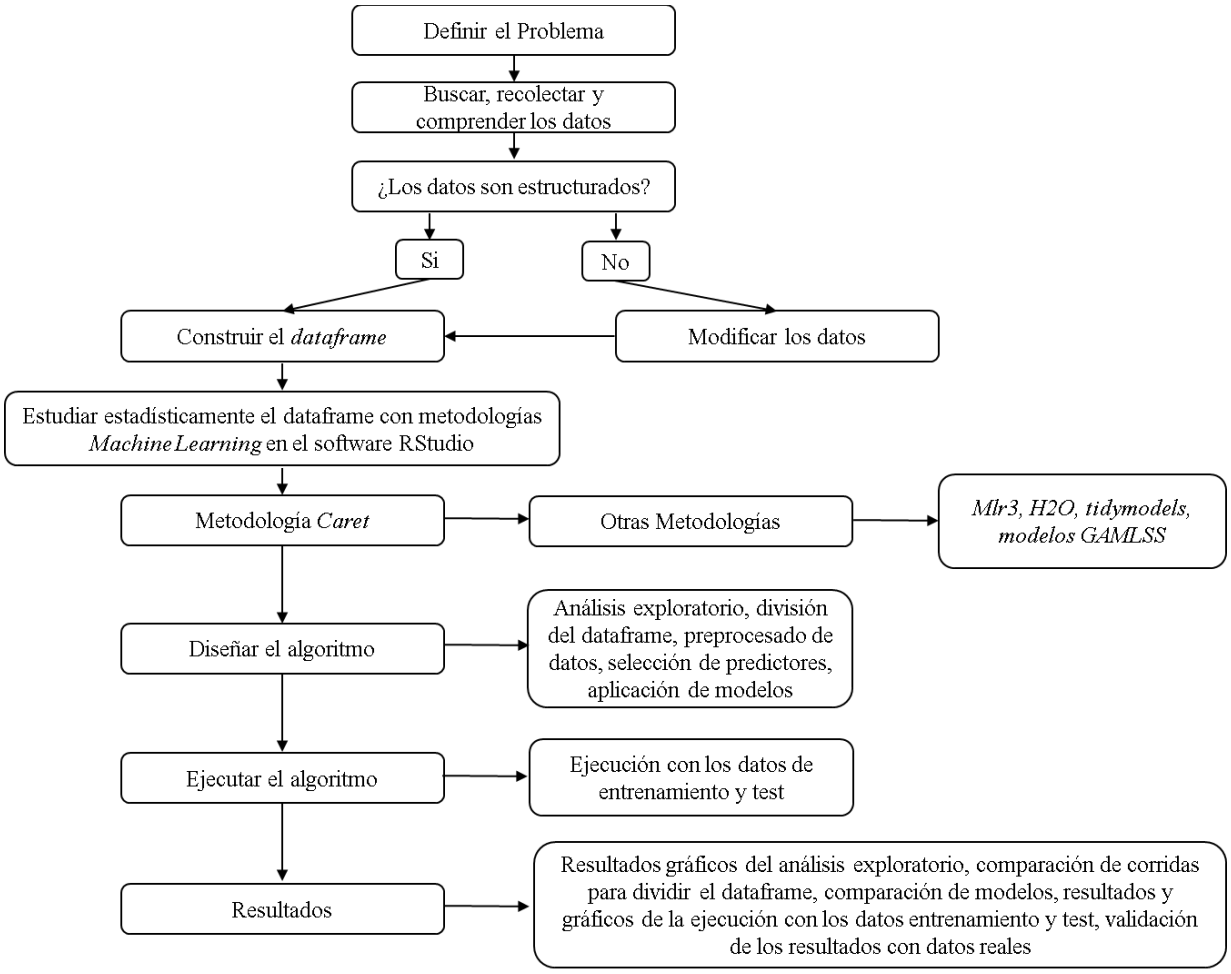
En este proyecto se planteó una metodología que permite desarrollar cada uno de los objetivos de forma ordenada y precisa. Esta se divide en dos secciones: metodología general y metodología para la aplicación de *Machine Learning* con el paquete *Caret*. Esta última se desarrolló haciendo uso del software RStudio y siguiendo los pasos planteados en el **ANEXO 2** La programación del algoritmo se elaboró ordenadamente y almacenando líneas de código por separado con el fin de optimizar las ejecuciones. Las funciones utilizadas para instalar cada uno de los paquetes fueron almacenadas en un archivo independiente, las cuales siempre deben ser cargadas con la función *library ( )* antes de ejecutar el algoritmo de *Machine Learning*, ya que estas no vienen cargadas por defecto. El algoritmo contiene alrededor de 2500 líneas de código, en las cuales se almacenan todos los códigos que permitieron tanto desarrollar el estudio estadístico como obtener los resultados de cada uno de los objetivos.

2.1. Metodología General

Esta sección abarca el conjunto de pasos desde la búsqueda de la información hasta la validación de los resultados finales, como se observa en la **Figura 5.**

Figura 5.

Diagrama de Flujo General del Proyecto



Nota. Diagrama de flujo explica el paso a paso que se siguió para dar solución a cada uno de los objetivos.

2.1.1. Definición del Problema

El punto de partida de este proyecto fueron los problemas que estaba presentando la *Compañía A* durante las operaciones de producción de hidrocarburos con equipos ESP. Se determinó que el comportamiento de estos equipos se veía afectado por parámetros de pozo, de yacimiento y de los fluidos, los cuales fueron denominados *Condiciones Especiales de Campo*. Es en este punto donde nació la idea de realizar un estudio estadístico utilizando los métodos modernos de computación estadística. Este estudio analizó en cada pozo el comportamiento de los equipos ESP de la *Compañía A* durante las operaciones de producción llevadas a cabo entre los años 2012 y 2020,

con el fin de mejorar la toma de decisiones sobre la aplicabilidad de estos equipos a partir de la predicción de una variable respuesta; en este caso fue una variable asociada a las fallas que han presentado los equipos en cada uno de los pozos, la que permitió predecir escenarios futuros.

2.1.2. Búsqueda, Recolección y comprensión de la Información

El primer paso fue generar una lista de los parámetros que podrían ser útiles para el estudio. Junto al director de la tesis, se logró establecer una lista que incluía parámetros del Equipo ESP, del pozo, del yacimiento y de los fluidos. Una vez terminada la lista, se inició la tarea de buscar la información contactando a los ingenieros de la *Compañía A* que estaban a cargo de cada cliente, para un total de 10 clientes. Una vez recolectada esta información, la cual fue suministrada bajo un acuerdo de confidencialidad, se inició la tarea de comprensión de los datos, donde el marco teórico juega un papel importante, ya que permitió entender los conceptos necesarios para determinar los tipos de datos, tipos de variables, datos estructurados y no estructurados, y datos correctos o incorrectos.

2.1.3. Construcción del Dataframe

Luego de comprender los datos que fueron recolectados, el siguiente paso fue construir una tabla de datos estructurada o *dataframe*, el cual quedó conformado por 51 variables (columnas) y 586 pozos (filas). Para poder construir el *dataframe*, primero se verificó que todos los datos fueran datos estructurados, es decir que su longitud, tamaño y formato estén definidos en formato tabla con filas y columnas con títulos. Los datos asociados a las categorías *Datos del Match*, *Datos de Producción* y *Condiciones Especiales de Campo* eran datos estructurados, pero no se encontraban en formato tabla, lo que generó un mayor trabajo de búsqueda y recolección. Los datos de *Cliente*, *Campo*, *Pozos*, categoría *Run Days*, categoría *Componentes del Equipo ESP* y categoría *Datos del Teardown* eran datos estructurados, que contaban con variables cualitativas que incluían nombres propios, las cuales fueron modificados por temas de confidencialidad. En la sección de resultados se incluye un gráfico que permite observar a detalle cada una de las variables que hacen parte del *dataframe* agrupadas según su origen.

2.1.4. Estudio estadístico con Machine Learning

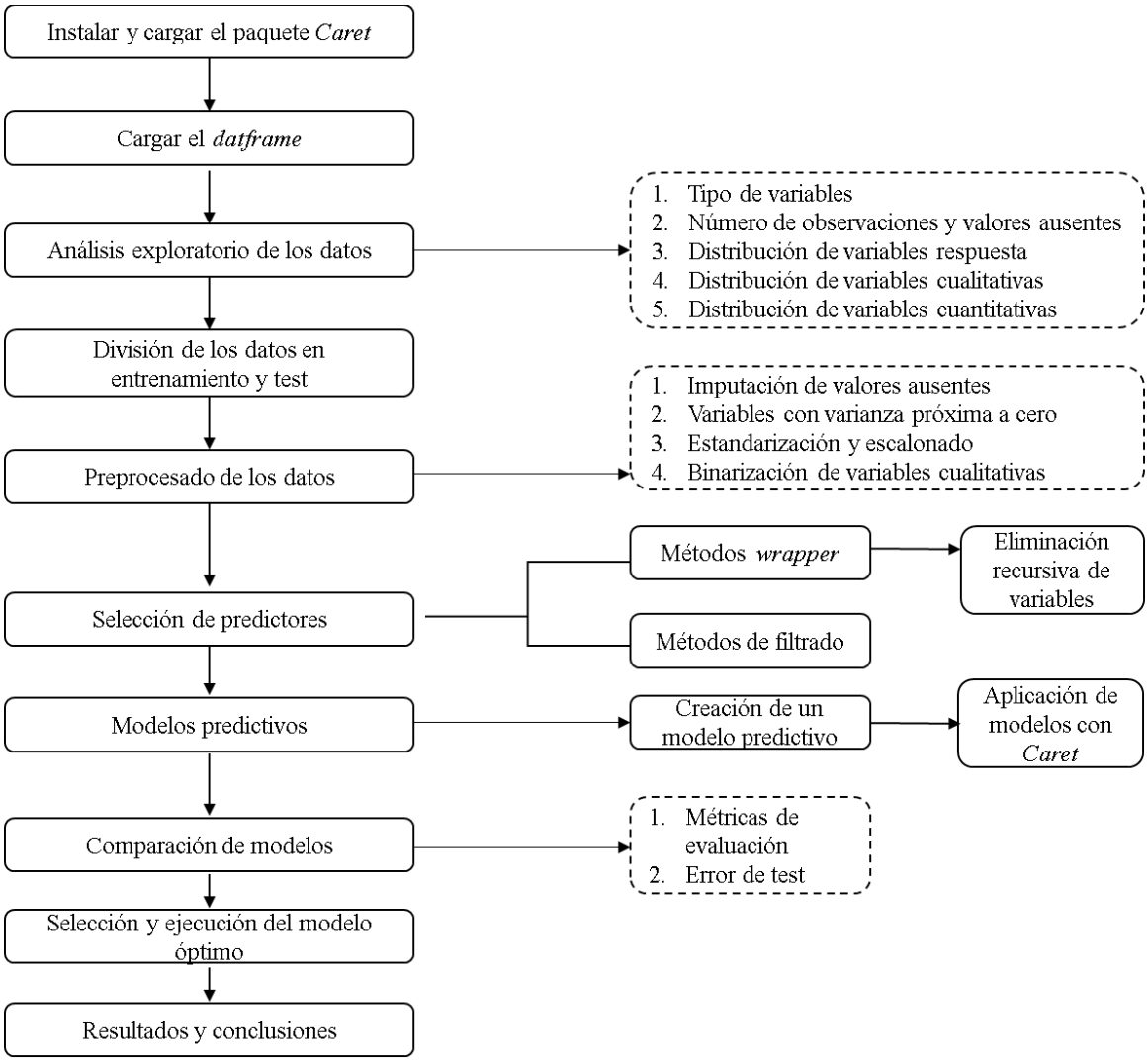
Para este estudio se indagó que metodologías de *Machine Learning* existen para desarrollarse con el software RStudio, entre las cuales se encontró que paquetes de RStudio como *Caret*, *mlr3*, *tidymodels* y *H2O* son utilizados para desarrollar algoritmos de *Machine Learning*. De estos se seleccionó el paquete *Caret* como la metodología para desarrollar el código, basándose del material formativo de Rodrigo [17]. El paso a paso que se llevó a cabo con este paquete se explica en la sección *Metodología Machine Learning con R y Caret*, junto con la explicación y ejecución del algoritmo diseñado.

2.2. Metodología Machine Learning con R y Caret

El algoritmo que se construyó para este proyecto está basado en la *Metodología de Machine Learning* del paquete ***Caret*** de RStudio. Este paquete fue desarrollado por *Max Kuhn*, y consiste en una interfaz que unifica bajo un único marco cientos de funciones de distintos paquetes, facilitando en gran medida todas las etapas de exploración, preprocesado, entrenamiento, optimización y validación de modelos predictivos [18]. La **Figura 6.**, recopila los pasos que se siguieron durante el desarrollo del algoritmo.

Figura 6.

Diagrama de Flujo para la Aplicación de la Metodología Caret



Nota. Diagrama de flujo que recolecta cada uno de los pasos que se siguieron para desarrollar el algoritmo haciendo uso de la metodología de Machine Learning con el paquete Caret de RStudio.

2.2.1. Carga del Dataframe

El primer paso fue cargar en el código el dataframe previamente ajustado, el cual se denominó "Dataframe_Principal". A partir de este se crearon nuevos dataframes modificados en el código, lo que permitió organizar los datos de tal forma que el análisis exploratorio fuera más preciso.

2.2.2. Análisis Exploratorio de los Datos

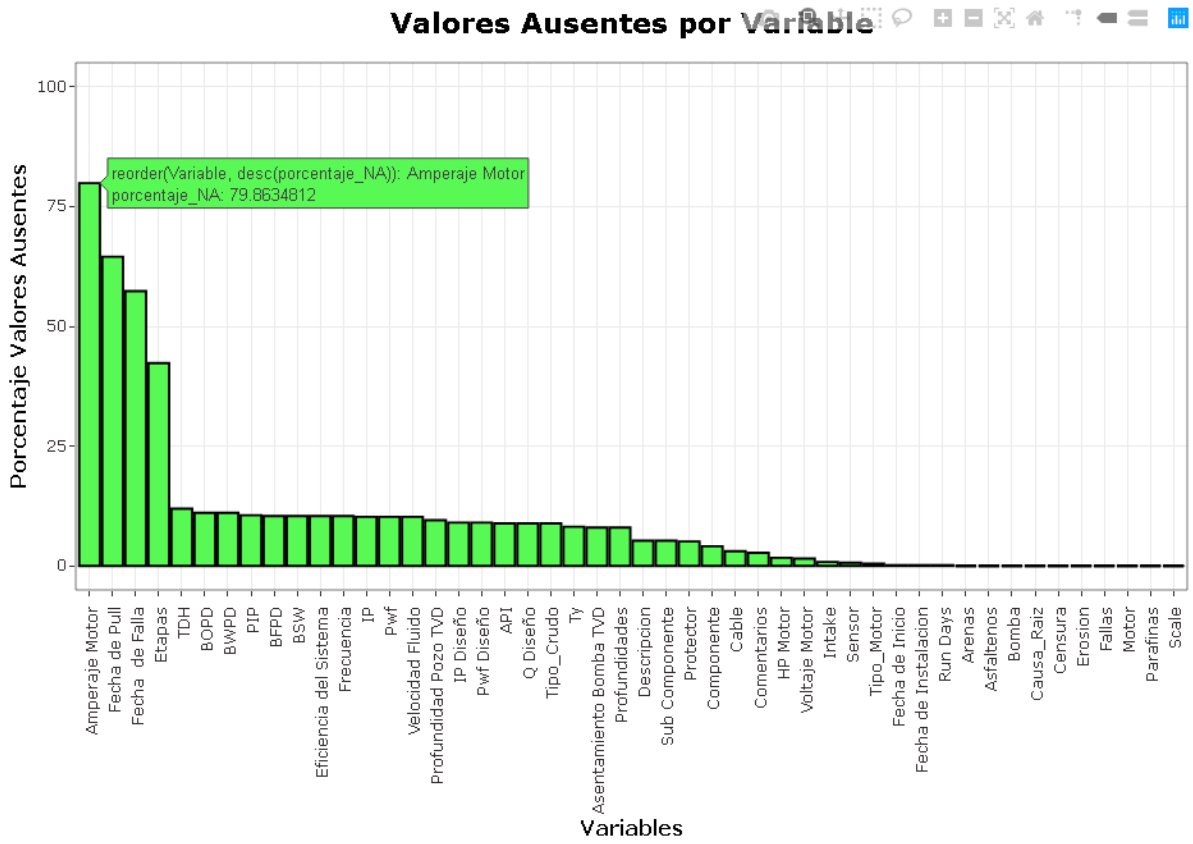
Este análisis consistió en explorar en cada uno de los valores que toma cada variable de cada pozo, agrupando los datos según su tipo en cualitativos nominales, cuantitativos discretos y continuos y valores ausentes (NA), con el fin de entender la información que se tenía; tipo de datos, cantidad de datos, cantidad de valores ausentes, distribución de variables cualitativas, distribución y correlación entre variables cuantitativas.

2.2.2.i. Tipo de variables. Se verificó que cada variable estuviera almacenada con el tipo de variable correcta, haciendo uso de la función *glimpse ( )*. De todas las variables solo el "IP Diseño" y el "IP" estaban almacenadas como caracteres, siendo variables numéricas. Esto se corrigió con la función *as.double ( )* que convierte cualquier variable en variable numérica.

2.2.2.ii. Número de Observaciones y Valores Ausentes. Con la función *nrow ( )* se obtuvo el número de renglones que en este caso es igual al número de pozos estudiados, para un total de 586. Llamando la función *map_dbl ( )* se conoció el número de datos que de 586 están ausentes por cada variable. Para observar esto gráficamente se programó un gráfico con la función *ggplot ( )* que muestra el porcentaje de valores ausentes por cada una de las 51 variables, que se observa en la **Figura 7.**

Figura 7.

Gráfico de Valores Ausentes

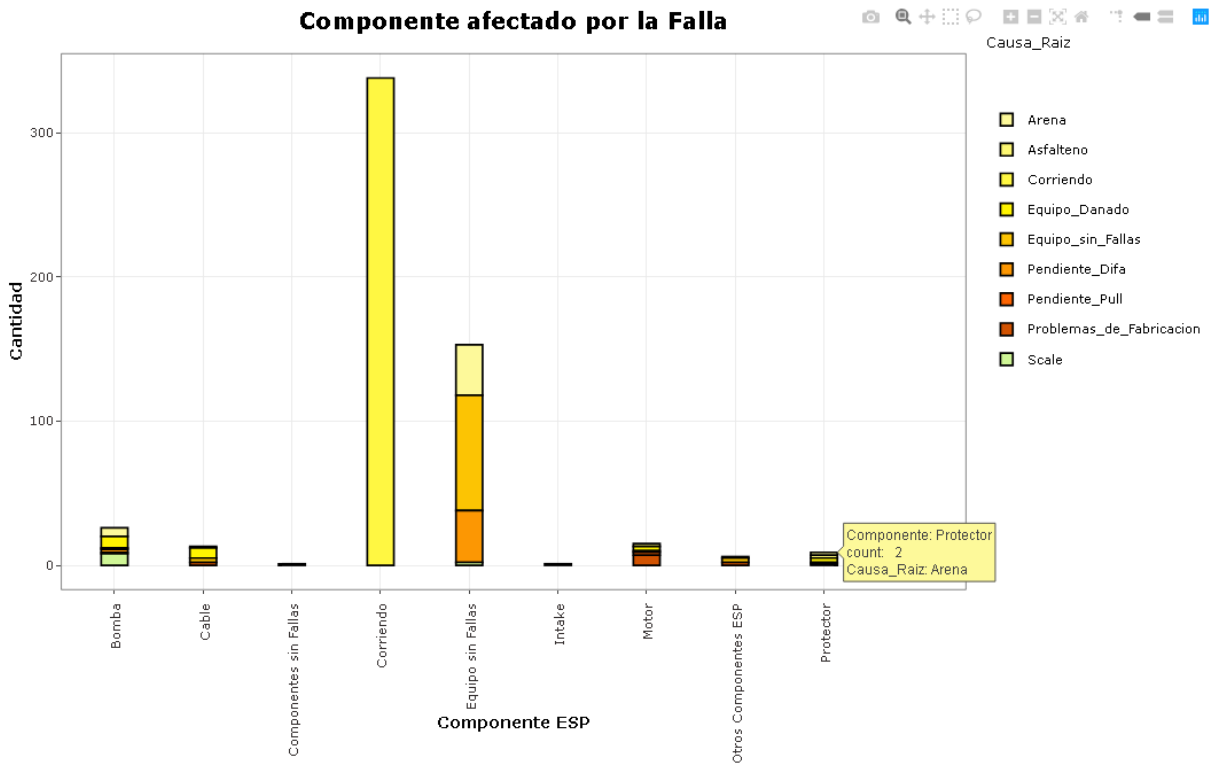


Nota. Gráfico que permitió tener una primera idea de cuáles son las variables más apropiadas para el estudio, siendo estas las que menor porcentaje de valores ausentes tienen.

2.2.2.iii. Distribución de Variables Respuesta. En este estudio se escogieron como variables respuesta aquellas variables dependientes que reflejan el resultado del comportamiento de los equipos ESP, como *Fallas, Comentarios, Causa Raíz, Descripción, Componente, Sub Componente y Censura*. Estas variables pertenecen a la categoría *Teardown*, y son resultados de los análisis realizados durante las operaciones de *Teardown*, que consiste en extraer y desarmar los equipos ESP que fallan durante las operaciones con el fin de determinar la causa raíz y el responsable. Siguiendo el mismo formato de la **Figura 8.**, en RStudio se programaron seis gráficos relacionando cada una de variables del grupo *Teardown* con la variable "Causa_Raiz", la cual se seleccionó como la variable respuesta a predecir debido a que menciona directamente las fallas que presentan los equipos ESP en cada pozo. Inicialmente, la variable "Causa_Raíz" tenía 28 niveles y varios de estos solo contaban con una repetición, lo que afectó considerablemente el ajuste del modelo, razón por la cual se modificaron algunos de los niveles asociando los niveles de otras variables respuesta como "Comentarios" y "Descripción". El total luego de esta transformación era de nueve niveles: ocho asociados a fallas y uno asociado a los equipos que están corriendo sin fallas. Estas modificaciones permitieron ajustar y ejecutar los modelos predictivos sin errores.

Figura 8.

Distribución de la Variable "Causa_Raiz"



Nota. En el recuadro inferior (amarillo claro) se observa que se tienen dos casos de fallas por Arena que afectaron directamente al Protector. Los niveles "Equipo_Dañado" y "Problemas de Fabricación" se refieren a los daños mecánicos que presentaron los equipos, con la diferencia que en el primero la responsabilidad es compartida entre el cliente y la Compañía A, y en el segundo solo la Compañía A. El nivel "Pendiente_Difa" significa que el equipo falló, se extrajo del pozo y se hizo el desarme, pero no se ha realizado el análisis de fallas. El nivel "Pendiente_Pull" significa que el equipo falló, pero no se ha realizado el desarme ni el análisis de fallas. El nivel "Equipo_sin_Fallas" hace referencia a los pozos que están parados por decisión del cliente. Debido a que no se podían dejar valores en blanco, se estableció el nivel "Corriendo" para los pozos que no presentan fallas. Los demás niveles hacen referencia a fallas por condiciones especiales de campo y son los más importantes a analizar en las predicciones.

Para complementar la información del gráfico anterior, se programó la **Tabla 4.**, para observar numéricamente como están distribuidos los niveles de la variable respuesta "Causa_Raiz" en el "Dataframe_Principal", es decir en el 100% de los datos.

Tabla 4.

Distribución de los Niveles de la Variable Respuesta

Niveles de Causa_Raiz	Distribución del Promedio de cada Nivel (%)
Arena	7.34
Asfalteno	1.02
Corriendo	57.68
Equipo Dañado	3.92
Equipo sin Fallas	16.38
Pendiente Difa	7.85
Pendiente Pull	1.71
Problemas de Fabricación	2.22
Scale	1.88

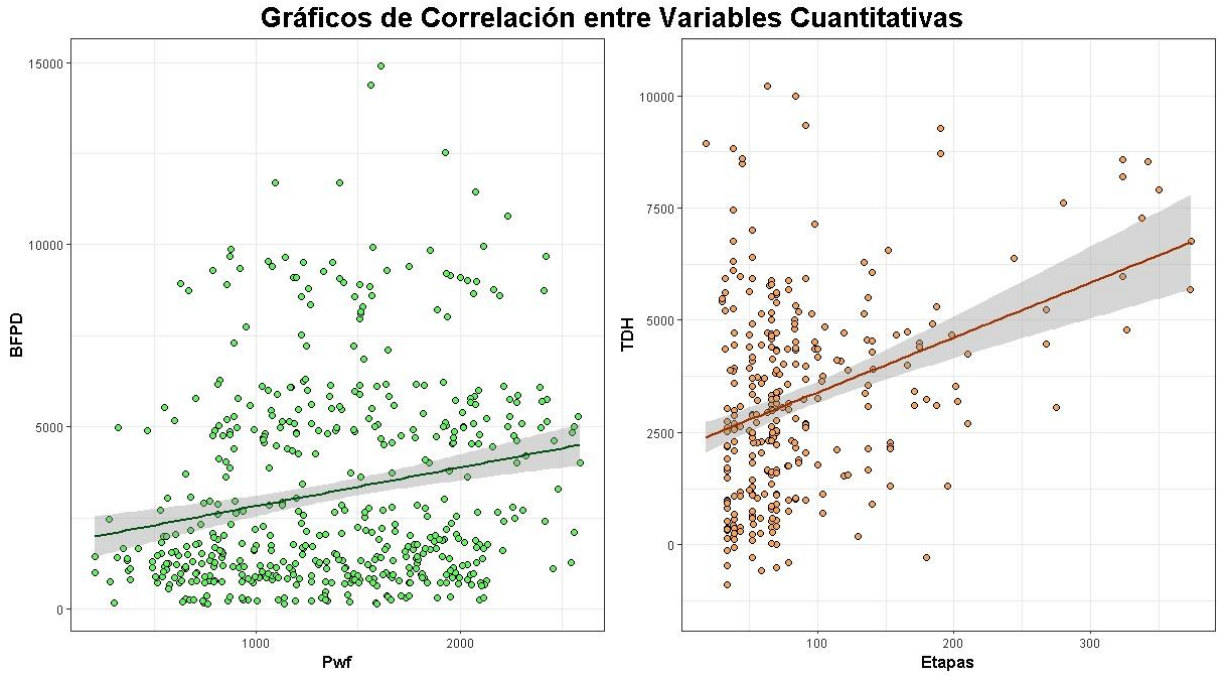
Nota. Valores que indican el promedio de la probabilidad de cada nivel calculada con el 100% de los datos.

2.2.2.iv. Distribución de Variables Cualitativas. En este proyecto se recolectaron un total de 14 variables cualitativas. Se programaron gráficos con el fin de observar cuantos niveles tenía cada una, y como se distribuía cada nivel con respecto a la variable respuesta "Causa_Raíz"; estos gráficos son mostrados en la sección de resultados debido a que permiten extraer información relevante para utilizar en el análisis de resultados. Las variables *Intake y Protector,* cuentan con más de 100 niveles; las variables *motor, cable, y sensor* cuentan con 50 o más niveles; la variable bomba tiene 29 niveles; las variables *Tipo_Motor, Profundidades y Tipo_Crudo* tienen menos de 10 niveles; y la variable *Parafinas* solo tiene un nivel, por lo cual se eliminó del estudio. Como se tenían muchos niveles, algunos de estos contenían pocas repeticiones cuando se analizaron con respecto a la "Causa_Raíz", lo que generó una distribución desequilibrada que afectó el código al momento de ajustar los modelos, razón por la cual se excluyeron las variables con más niveles al momento de programar el preprocesado de datos. De las variables "Arena", "Asfaltenos", "Erosión" y "Scale", no se contaba con los valores numéricos, por lo cual fue necesario estimarlos como valores cualitativos binarios ("Si" y "No") a partir de la información de la variable "Comentarios"; esto puede generar predicciones inesperadas.

2.2.2.v. Distribución de Variables Cuantitativas. En primer lugar, se analizó la relación entre las variables cuantitativas (entre discretas y continuas suman 22 variables) y la variable "Causa_Raíz". Se realizaron gráficos de densidad y gráficos de caja, haciendo uso de las funciones *geom_density ( )* y *geom_boxplot ( )* respectivamente, los cuales permitieron observar cómo se distribuían los valores que toman estas variables en función de los niveles de la "Causa_Raíz". Los gráficos anteriores fueron útiles para observar distribuciones, sin embargo, se hizo mayor énfasis en las correlaciones que existe entre estas variables cuantitativas, ya que estas permitieron escoger solo las variables más representativas evitando que se añadiera información redundante al modelo, al seleccionar variables que estuvieran mutuamente correlacionadas. Esto se logró haciendo uso de funciones de R que permitieron programar gráficos y test de correlación basados en el "Coeficiente de correlación de Pearson"; una medida de dependencia lineal entre dos variables cuantitativas. Primero se programaron gráficos de correlación por cada pareja de variables cuantitativas como se observa en la **Figura 9.**, utilizando las funciones *geom_point ( )* y *geom_smooth (..., method = "lm")*, esta última grafica una recta que muestra la distribución lineal que tienen los puntos graficados.

Figura 9.

Gráficos de Correlación



Nota. En el gráfico de la izquierda se graficó BFPD vs Pwf y se observó que los puntos están muy dispersos con respecto a recta que representa la distribución lineal de Pearson, la cual define que existe cierta relación directa entre ambas variables. En el gráfico de la derecha se graficó TDH vs Etapas y se observó que la dispersión de los puntos es más ordenada con respecto a la recta de distribución lineal y de igual forma se tiene una correlación directamente proporcional.

Visualmente esto se sustenta con el gráfico que se muestra en la sección de resultados conocido como *Correlograma*, el cual consiste en una matriz de correlación que permite analizar la relación entre cada par de variables numéricas de un conjunto de datos, agrupando diagramas de dispersión, diagramas de distribución de variable y coeficiente de correlación de Pearson [19].

El análisis exploratorio permitió realizar el primer filtro de 25 variables, entre cualitativas, cuantitativas, cualitativas y variables respuesta. Con estas se programaron las funciones de preprocesado de datos y de métodos de selección, con el objetivo de seleccionar definitivamente cuales de estas variables serían las más relevantes para ejecutar la predicción.

2.2.3. División del Dataframe en Entrenamiento y Test

El "Dataframe_Principal" se dividió en dos partes: "Dataframe_Entrenamiento" y "Dataframe_Test". El primero de estos se utilizó para ajustar y entrenar cada uno de los modelos

predictivos programados, y el segundo para evaluar y validar la capacidad predictiva de los mismos. La división se realizó de forma aleatoria con la función *set.seed (350)*, y se programó con la función (y su argumento) *createDataPartition (y = Dataframe_Principal$`CausaRaiz`, p=0.8…)* donde el argumento *p = 0.8* hace referencia a que el conjunto de entrenamiento será un 80% del "Dataframe_Principal", y el conjunto *test* será el 20% restante. Esta división se hizo en función de la variable respuesta "Causa_Raíz" ya que al ser esta la variable a predecir debe estar distribuida equitativamente en ambos conjuntos; esto se comprobó programando la función *prop.table (table (...))* a cada uno de los nuevos conjuntos, y evaluando varios escenarios de división que se muestran en los resultados de este proyecto. Es importante aclarar que en los siguientes pasos el conjunto de test no participó de ningún ajuste de los modelos.

2.2.4. Preprocesado de los Datos

Este paso fue de gran importancia ya que permitió transformar los datos que tiene el "Dataframe_Principal" con la finalidad de que fueran aceptados por los algoritmos de *Machine Learning* del paquete *Caret*. Para este proceso se creó un objeto llamado "Objeto_Recipe" el cual contiene la función *récipe ( )* y como argumentos tiene definido la variable respuesta, los predictores (los seleccionados en el análisis exploratorio) y el conjunto de datos "Dataframe_Entrenamiento". Todas las transformaciones que se realizaron fueron almacenadas en este objeto con el fin de poder aplicarlas al "Dataframe_Principal".

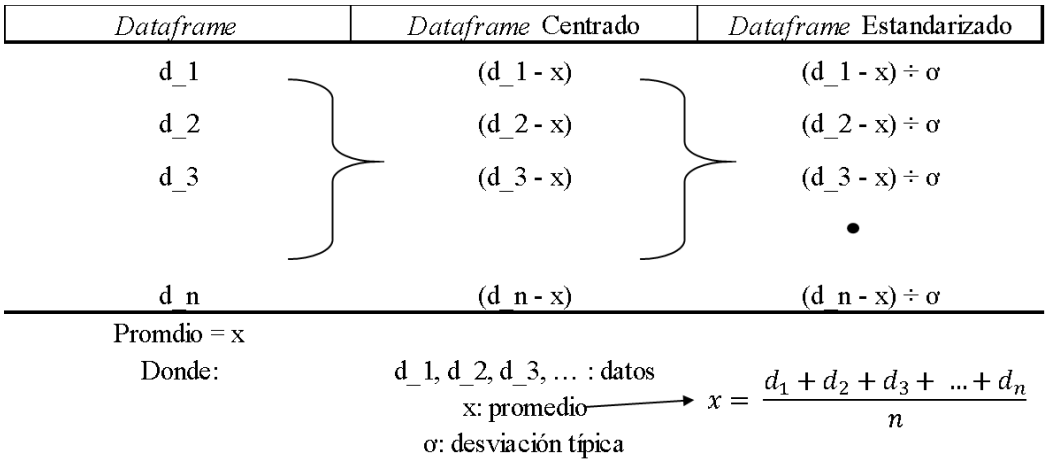
2.2.4.i. Imputación de Valores Ausentes. Si se observa la **Figura 7.**, la cantidad de valores ausentes (NA) no es significativa, sin embargo, se debe tener precaución ya que la gran mayoría de algoritmos de *Machine Learning* no aceptan estos valores, razón por la cual se deben transformar ya sea eliminando los renglones que cuentan con valores NA, eliminando las variables que contengan valores NA, o tratando de estimar los valores NA. Esta última opción fue la que se aplicó utilizando la función *step_bagimpute ( )* y como argumento las variables a imputar; de esta forma los valores NA son estimados a partir del resto de la información disponible.

2.2.4.ii. Valores con Varianza Próxima a Cero. Las variables que contengan un único valor o una variación mínima no aportan información al modelo predictivo, por ende, no incluyen como predictores. Para comprobar esto se programó la función *step_nzv ( )* y como argumento las variables cuantitativas, y se almacenó en el "Objeto_Recipe". Se determinó que ninguna variable tenía varianza cero o próxima a cero.

2.2.4.iii. Estandarización y Escalonado. A cada una de las variables cuantitativas se les realizó este proceso que consistió en buscar la manera de igualar tanto la escala como la magnitud de estas. Esto se hizo con el fin de evitar que en el modelo predictivo no dominen las variables que tiene valores más altos, sino las variables que realmente tiene mayor influencia. La estrategia consistió primero en centrar los valores y luego en estandarizarlos, como se explica en la **Figura 10.** La centralización se programó con la función *step_center (all_numeric ( ))* y la estandarización con la función *step_scale (all_numeric ( ))*, e igualmente se almacenaron en el "Objeto_Recipe".

Figura 10.

Estandarización y Escalonado



Nota. Explicación de las fórmulas que utilizan las funciones step_center ( ) y step_scale ( ) para transformar los valores numéricos. Tomado de: CienciadeDatos.net. (Abril, 2018). "Machine Learning con R y Caret" [En línea]. https://www.cienciadedatos.net/documentos/41_machine_learning_con_r_y_caret. [Acceso: septiembre 10, 2020]. Modificado por J.S. Andrade.

2.2.4.iv. Binarización de Variables Cualitativas. Este paso consistió en crear variables *dummy*, es decir, variables creadas a partir de los niveles de una variable cualitativa que pueden tomar dos valores: 0 y 1, que indican ausencia y presencia de dicha variable cualitativa en cada renglón. Por ejemplo, la variable "Tipo_Crudo" tiene cuatro niveles: extrapesado, pesado, mediano y liviano, y su transformación quedó "Tipo_Crudo.extrapesado", "Tipo_Crudo.pesado", y así sucesivamente. De igual forma sucederá con cada variable cualitativa que se agregue como argumento a la función *step_dummy ( )*. Se debe tener precaución de que variables se van a agregar, ya que si contienen muchos niveles el tiempo de corrida será indefinido.

Una vez se finalizó cada uno de estos pasos y luego de que todos se almacenaran en el "Objeto_Recipe", el siguiente paso fue entrenar dicho objeto con el conjunto de datos de entrenamiento y luego se aplicó el entrenamiento tanto al "Dataframe_Entrenamiento" como al "Dataframe_Test" con la función *bake ( )*. La **Figura 11.**, muestra el resumen de las variables y sus respectivos valores, de los nuevos dataframes que fueron el resultado de cada transformación, los cuales fueron almacenados como "Dataframe_Entrenamiento_Preprocesado" y

"Dataframe_Test_Preprocesado" para identificarlos fácilmente en los siguientes pasos de la metodología.

Figura 11.

Resumen de los Resultados del Preprocesado

```
Rows: 472
Columns: 26
$ Run_Days                <dbl> -0.71595617, -1.12676154, -0.33369619, -0.58812248, -0.81028006, -0.07182328, -0.78918130, -0.01...
$ Bomba                   <fct> BOMBA 23, BOMBA 26, BOMBA 26, BOMBA 27, BOMBA 25, BOMBA 28, BOMBA 26, BOMBA 12, BOMBA 14, BOMBA ...
$ Etapas                  <dbl> 1.77541640, -0.70555200, -0.70555200, 0.52637714, 1.27922272, 0.85146955, 0.80013917, 2.11761893...
$ Motor                   <fct> MOTOR 34, MOTOR 28, MOTOR 28, MOTOR 30, MOTOR 35, MOTOR 27, MOTOR 36, MOTOR 13, MOTOR 19, MOTOR ...
$ Tipo_Motor              <fct> TIPO 6, TIPO 4, TIPO 4, TIPO 3, TIPO 7, TIPO 1, TIPO 6, TIPO 3, TIPO 3, TIPO 3, TIPO 2, TIPO 2, ...
$ HP_Motor                <dbl> 0.472340534, -0.002071117, -0.002071117, 1.726142755, 0.641773266, -0.103730757, 0.811205999, -0...
$ Asentamiento_Bomba_TVD  <dbl> 2.61136472, 2.45605312, 2.45605312, -0.15600783, -0.24407984, -0.02148026, -0.21698076, -0.25085...
$ Velocidad_Fluido        <dbl> -0.59411415, -1.28263725, -1.21079136, 1.11521929, 0.38777967, -1.55205933, -0.15684281, -0.8276...
$ Frecuencia              <dbl> 0.562316020, 1.704109418, -0.157374911, -0.440644266, 0.119668743, -0.627415269, -0.552337952, -...
$ Eficiencia_del_Sistema  <dbl> 1.317060540, 0.398229480, 0.908691180, -0.877924769, 1.470199050, 0.806598840, -0.003961325, -1....
$ PIP                     <dbl> -0.90564168, -0.72603187, 0.37409064, -0.87041275, 1.42574101, -1.32585972, 0.15046916, -1.32844...
$ BFPD                    <dbl> -0.77483583, 0.12099056, 0.13406574, 0.54575716, -0.57906154, -0.67977575, 0.04702132, -0.311903...
$ BSW                     <dbl> -0.994773579, 1.417896419, 0.101894602, 1.198562783, 0.321228238, 0.880529011, -0.231977653, 1.0...
$ TDH                     <dbl> 2.87629346, 2.52223935, 2.56480907, 0.52970806, -0.38035334, 0.74459194, -0.28409400, 0.59985853...
$ Ty                      <dbl> 2.837605330, 2.501439207, 2.501439207, 0.484442466, 0.964679785, 1.204798445, 0.964679785, 0.964...
$ IP                      <dbl> 0.75665473, 0.55926812, 0.38655483, -0.39682328, -0.13158502, -0.98281478, -0.08217088, -0.78954...
$ API                     <dbl> 2.110966754, 1.658131272, 1.658131272, 0.027923539, 0.088301603, 0.027923539, 0.088301603, -0.00...
$ Tipo_Crudo              <fct> Liviano, Liviano, Liviano, Pesado, Pesado, Pesado, Pesado, Pesado, Pesado, Pesado, Pesado, Pesad...
$ Causa_Raiz              <fct> Arena, Equipo_Danado, Corriendo, Corriendo, Corriendo, Corriendo, Corriendo, Corriendo, Corriend...
$ Profundidades_Profundo  <dbl> 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, NA, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0,...
$ Profundidades_Somero    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, NA, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
$ Arenas_Si               <dbl> 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
$ Asfaltenos_Si           <dbl> 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
$ Erosion_Si              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
$ Scale_Si                <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
$ Fallas_Si               <dbl> 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, ...
```

Nota. Con la función glimpse (Dataframe_Entrenamiento_Preprocesado) se obtuvo este resultado que resume como quedaron los datos de entrenamiento y de test luego de la transformación realizada durante el preprocesado.

2.2.5. Selección de Predictores

Como se ha mencionado anteriormente, antes de crear el modelo predictivo debo seleccionar como predictores únicamente las variables que están realmente relacionadas con la variable respuesta "Causa_Raíz"; incluir un exceso de variables suele conllevar a una reducción de la capacidad predictiva del modelo cuando se expone a nuevos datos [18]. Existen varios métodos que facilitan este proceso, como los Métodos *Wrapper* y lo Métodos de Filtrado, los cuales fueron los que se utilizaron en este proyecto.

2.2.5.i. Métodos Wrapper. El paquete *Caret* incorpora métodos *wrapper* basados en eliminación recursiva, algoritmos genéticos y *simulated annealing* [18]. En este proyecto se implementó la eliminación recursiva de variables, la cual es una estrategia muy práctica para evitar la búsqueda exhaustiva. La idea detrás del algoritmo que se programó para aplicar este método puede resumirse en los siguientes pasos: primero se escogió el tamaño de los conjuntos predictores y el número de repeticiones para el proceso de validación cruzada (proceso que evita que en el ajuste se incorporen datos del "Dataframe_Test_Preprocesado"), luego se programó el bucle de validación cruzada y el control de entrenamiento donde se definió el tipo de modelo empleado para la selección (modelo *rfFuncs* que se refiere a un modelo *Random Forest*), y por último se ejecutó la eliminación recursiva de predictores utilizando la función *rfe ( )*. El criterio de selección fue escoger el número de predictores que mostraron mayor exactitud según el gráfico que se muestra en la sección de resultados, ya que, según el principio de parsimonia, entre un conjunto de modelos con la misma capacidad predictiva, el mejor es el más simple [18].

2.2.5.ii. Métodos de Filtrado. Este método evalúa la relevancia de los predictores fuera del modelo, para posteriormente, incluir únicamente aquellos que pasen determinado criterio [18]. Para programar este método se siguió el mismo resumen de pasos del método anterior, variando el tipo de modelo para el control de entrenamiento por *rfSBF*, y el paso final donde se ejecutó el filtrado utilizando la función *sbf()*.

En la **Figura 12.**, se comparan las variables óptimas que resultaron de cada método de selección, donde se observa la cantidad de variables que arrojó el método de filtrado con respecto al método de eliminación recursiva de variables, que da una primera idea de cuál es el método más preciso para escoger los predictores de un modelo. Todas las modificaciones que se le realizaron a los *dataframes* preprocesados se almacenaron en dos variables: "Dataframe_Entrenamiento_Modelo" y "Dataframe_Test_Modelo", las cuales se crearon a partir de los predictores más importantes seleccionados en el método de eliminación recursiva de variables, los cuales se utilizaron para crear los modelos predictivos.

Figura 12.

Resultados de las Variables Óptimas

Método de eliminación recursiva de variables

```
> Rf_rfe$optVariables
 [1] "Fallas_Si"          "Run_Days"           "BFPD"               "Asentamiento_Bomba_TVD" "Arenas_Si"
 [6] "API"                "TDH"                "HP_Motor"           "Ty"                     "Etapas"
[11] "BSW"                "Frecuencia"         "PIP"                "IP"                     "Eficiencia_del_Sistema"
[16] "Velocidad_Fluido"   "Tipo_CrudoPesado"   "Erosion_Si"         "Tipo_MotorTIP0 7"       "Profundidades_Somero"
```

Método de Filtrado

```
> rf_sbf$optVariables
 [1] "Run_Days"           "BombaB0MBA 13"      "BombaB0MBA 17"      "BombaB0MBA 18"      "BombaB0MBA 2"
 [6] "BombaB0MBA 23"      "BombaB0MBA 26"      "BombaB0MBA 27"      "BombaB0MBA 28"      "BombaB0MBA 4"
[11] "BombaB0MBA 5"       "MotorM0T0R 16"      "MotorM0T0R 17"      "MotorM0T0R 2"       "MotorM0T0R 27"
[16] "MotorM0T0R 28"      "MotorM0T0R 36"      "MotorM0T0R 44"      "MotorM0T0R 45"      "MotorM0T0R 46"
[21] "MotorM0T0R 48"      "MotorM0T0R 5"       "MotorM0T0R 6"       "Tipo_MotorTIP0 2"   "Tipo_MotorTIP0 3"
[26] "Tipo_MotorTIP0 6"   "HP_Motor"           "Asentamiento_Bomba_TVD" "BFPD"           "BSW"
[31] "TDH"                "Ty"                 "API"                "Tipo_CrudoLiviano"  "Tipo_CrudoMediano"
[36] "Tipo_CrudoPesado"   "Profundidades_Profundo" "Profundidades_Somero" "Arenas_Si"      "Asfaltenos_Si"
[41] "Erosion_Si"         "Scale_Si"           "Fallas_Si"
```

Nota. Luego de haber corrido cada método, se llama a cada uno con "$optVariables" con el fin de obtener estos resultados. Las variables se seleccionan en la sección de resultados según el número de predictores obtenido de la gráfica de la **Figura 25.**

2.2.6. Modelos Predictivos

Después del preprocesado y la selección de predictores, el siguiente paso fue emplear un algoritmo de *Machine Learning* que permitiera crear un modelo capaz de representar los patrones presentes en los datos del "Dataframe_Entrenamiento_Modelo", para luego poder emplearlo para hacer la predicción con los datos que hasta el momento no han participado del ajuste, el "Dataframe_Test_Modelo". Existen multitud de algoritmos de *Machine Learning*, cada uno con características propias y con distintos parámetros que deben ser ajustados, lo que dificulta la tarea de encontrar el mejor modelo. Algunos de los modelos que hacen parte del paquete *Caret* son [18]: K-Nearest Neighbor (KNN), Naive Bayes, Regresión Logística, LDA, Árbol de Clasificación Simple, Random Forest, Gradient Boosting, SVM, y Redes Neuronales (NNET). En general, cada uno de estos modelos fue programado siguiendo las etapas de la **Tabla 5.**

Tabla 5.

Etapas de un Modelo Predictivo

ETAPA	EXPLICACIÓN
1. Ajuste/Entrenamiento	Se aplicó un algoritmo de *Machine Learning* a los datos de "Dataframe_Entrenamiento_Modelo" para que el modelo aprendiera a partir del entrenamiento.
2. Evaluación/Validación	El objetivo de la evaluación del modelo no es ser capaz de predecir observaciones que ya se conocen, sino nuevas observaciones que el modelo nunca ha visto. Para poder estimar el error que cometió el modelo se incluyeron líneas de código que aplican estrategia de validación, como la validación cruzada.
3. Optimización de Hiperparámetros	Los hiperparámetros son aquellos parámetros de un algoritmo de *Machine Learning* que no se aprenden con los datos, sino que son el resultado de configuraciones realizadas durante el entrenamiento del modelo. Para conocer el valor exacto de un hiperparámetro, se aplicaron diferentes estrategias según cada modelo con el fin de observar dentro de un rango de valores, cuál era el valor que arrojaba el mejor resultado.
4. Predicción	Una vez se creó cada modelo y se entrenaron con el conjunto de datos de entrenamiento, este se empleó para predecir nuevas observaciones, usando los datos del "Dataframe_Test_Modelo".

Nota. Para cada uno de los modelos predictivos aplicados en este proyecto se siguieron estas etapas de forma ordenada durante la programación de estos en RStudio. Tomado de: CienciadeDatos.net. (Abril, 2018). "Machine Learning con R y Caret" [En línea]. https://www.cienciadedatos.net/documentos/41_machine_learning_con_r_y_caret. [Acceso: septiembre 10, 2020]. Modificado por J.S. Andrade.

Para este proyecto se programaron cinco modelos siguiendo la estructura del algoritmo mostrado en la **Figura 13.**, con el fin de exponer escenarios que permitieran seleccionar el modelo óptimo.

Figura 13.

Algoritmo programado en RStudio para los Modelos Predictivos

A. Ajustes iniciales

```
# 1. Paralelización, Repeticiones, Hiperparámetros y Semillas
# ============================================================================================

registerDoParallel(cores = 4)
particiones <- 10
repeticiones <- 5

Hiperparametros <- expand.grid(mtry = c(3,4,5,7),
                               min.node.size = c(2,3,4,5,10,15,20,30),
                               splitrule = "gini")

set.seed(123)
Seeds <- vector(mode="list", length = (particiones * repeticiones) +1)
for(i in 1:(particiones * repeticiones)){
  Seeds[[i]] <- sample.int(1000, nrow(Hiperparametros))
}
Seeds [[(particiones * repeticiones) + 1]] <- sample.int(1000,1)
```

B. Definición y ejecución del modelo

```
# 2. Definición del entrenamiento
# ============================================================================================

Control_Entrenamiento <- trainControl(method = "repeatedcv", number = particiones,
                                      repeats = repeticiones, seeds = Seeds,
                                      returnResamp = "final", verboseIter = FALSE,
                                      classProbs = TRUE, allowParallel = TRUE )

# 3. Ajuste del modelo
# ============================================================================================

set.seed(342)
Modelo_RF<- train(Causa_Raiz ~.,
                  data = Dataframe_Entrenamiento_Modelo,
                  method = "ranger",
                  tuneGrid = Hiperparametros,
                  metric = "Accuracy",
                  trControl = Control_Entrenamiento,
                  num.trees = 500)
Modelo_RF$finalModel
```

C. Predicción y representación gráfica de la exactitud

```
# 4. Predicción
# ============================================================================================

Predicciones_RF_raw <- predict(Modelo_RF, newdata = Dataframe_Test_Modelo, type = "raw")
Predicciones_RF_raw

Predicciones_RF_prob <- predict(Modelo_RF, newdata = Dataframe_Test_Modelo, type = "prob")
Predicciones_RF_prob


# 5. Representación Gráfica
# ============================================================================================

Grafico_33 <- ggplot(Modelo_RF, highlight = TRUE)+
              scale_x_continuous(breaks = 1:30)+
              labs(title = "Evolución de la Exactitud del Modelo Random Forest",
                   x = "# mínimo de nodos",
                   y = "Exactitud con Validación Cruzada") +
              guides(color = guide_legend(title = "Hiperparámetros"),
                     shape = guide_legend(title = "mtry"))+
              theme_bw()+
              Estetica_Ejes_2
```

Nota. Estructura que se programó para cada uno de los cinco modelos, variando cuidadosamente los hiperparámetros, los métodos aplicados y los nombres de los objetos donde se almacena toda la información, sin variar el número de semillas aleatorias.

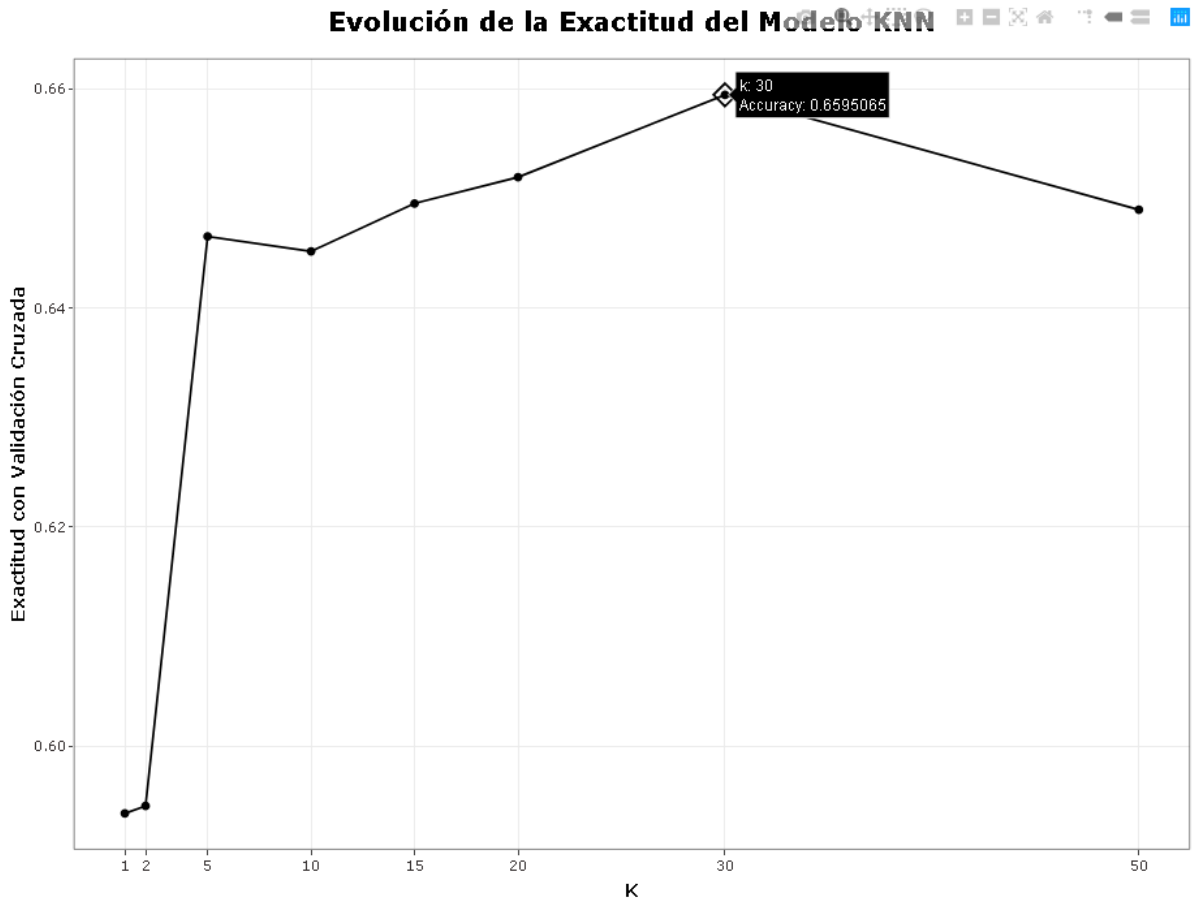
De las ejecuciones de cada uno de los modelos que se explican a continuación, el resultado más importante que se obtuvo en cada caso fue el gráfico que representa el número de hiperparámetros, los cuales fueron analizados rigurosamente con el fin de mostrar cual era el número de hiperparámetros con el que se lograba el valor de exactitud más alto; estos gráficos permitieron obtener una primera idea de cuál podría ser el modelo óptimo para seleccionar. Estas ejecuciones se realizaron con el conjunto de datos "Dataframe_Entrenamiento_Modelo", y los resultados mostrados a continuación solo hacen referencia a información propia de cada modelo que permite entender, comparar y seleccionar el modelo óptimo. La extracción de resultados asociados a las predicciones obtenidas durante el entrenamiento se muestra en la sección de resultados.

2.2.6.i. Modelo Árbol de Clasificación Simple. Consiste en un árbol de clasificación que fue desarrollado como un subtipo de los árboles de regresión, pero con el objetivo de clasificar variables cualitativas [20]. Para programar este modelo se utilizó la función *C5.0.default ( )* del paquete C50 de *Caret*, y no fue necesario ajustar ningún hiperparámetro, ya que este modelo no lo tiene en cuenta dentro de sus cálculos internos. Esto generó que no fuera posible graficar la relación entre el valor del hiperparámetro y la exactitud del modelo que se graficó con los otros modelos, razón por la cual se anticipó un rechazo de este modelo, por ende, no se muestra la evidencia respectiva.

2.2.6.ii. Modelo KNN. El modelo *K-Nearest Neighbor* es uno de los algoritmos de *Machine Learning* más simples. Se fundamenta en la idea de identificar observaciones en el conjunto de entrenamiento que se asemejen a la observación de test, y asignarle como valor predicho la clase predominante entre dichas observaciones; a pesar de su sencillez, en muchos escenarios consigue resultados aceptables [18]. Para este modelo se empleó la función *knn3 ( )* y solo se ajustó el hiperparámetro *k*, el cual toma valores a partir de observaciones vecinas. Como resultado de la ejecución se obtuvo **Figura 14.**

Figura 14.

Gráfico de Exactitud KNN

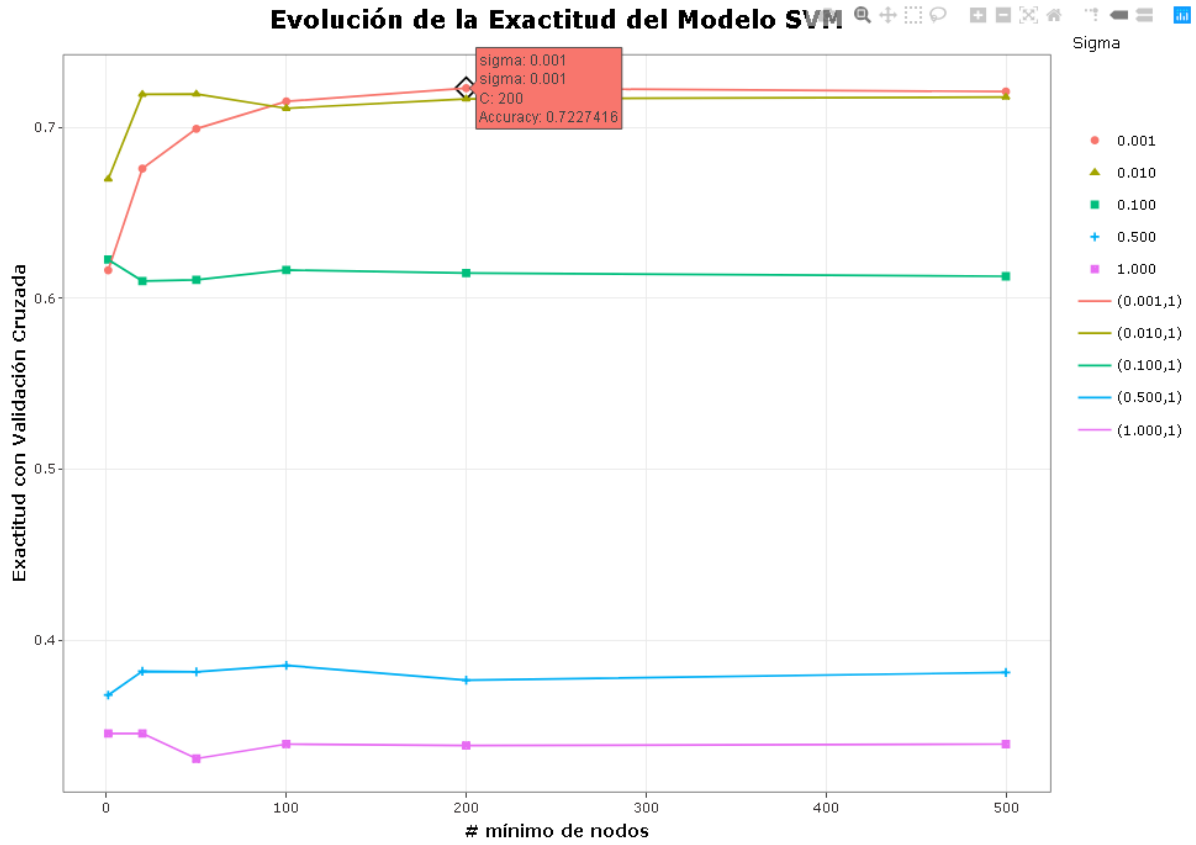


Nota. El eje x toma los valores del hiperparámetro k y el eje y los valores de exactitud que se logran con cada uno de los valores de k. El valor óptimo del hiperparámetro (k = 30) alcanzó una exactitud de 65.95%, un valor bajo debido a la sencillez del modelo de solo utilizar un hiperparámetro.

2.2.6.iii. Modelo SVM. El modelo de clasificación *Máquinas de Vector Soporte* (*Suport Vector Machines, SVM*) inicialmente fue desarrollado para métodos de clasificación binaria, y hoy en día son ampliamente utilizados en problemas de clasificación múltiple y de regresión. Con la función *ksvm ( )* del paquete *kernlab* de *Caret* se hicieron los ajustes a la estructura de código general. En este caso se ajustaron dos hiperparámetros: *sigma* como el coeficiente radial de Kernel, y *C* como penalización por violaciones del margen del hiperplano [18]. La **Figura 15.**, muestra los resultados de la ejecución de este modelo.

Figura 15.

Gráfico de Exactitud SVM

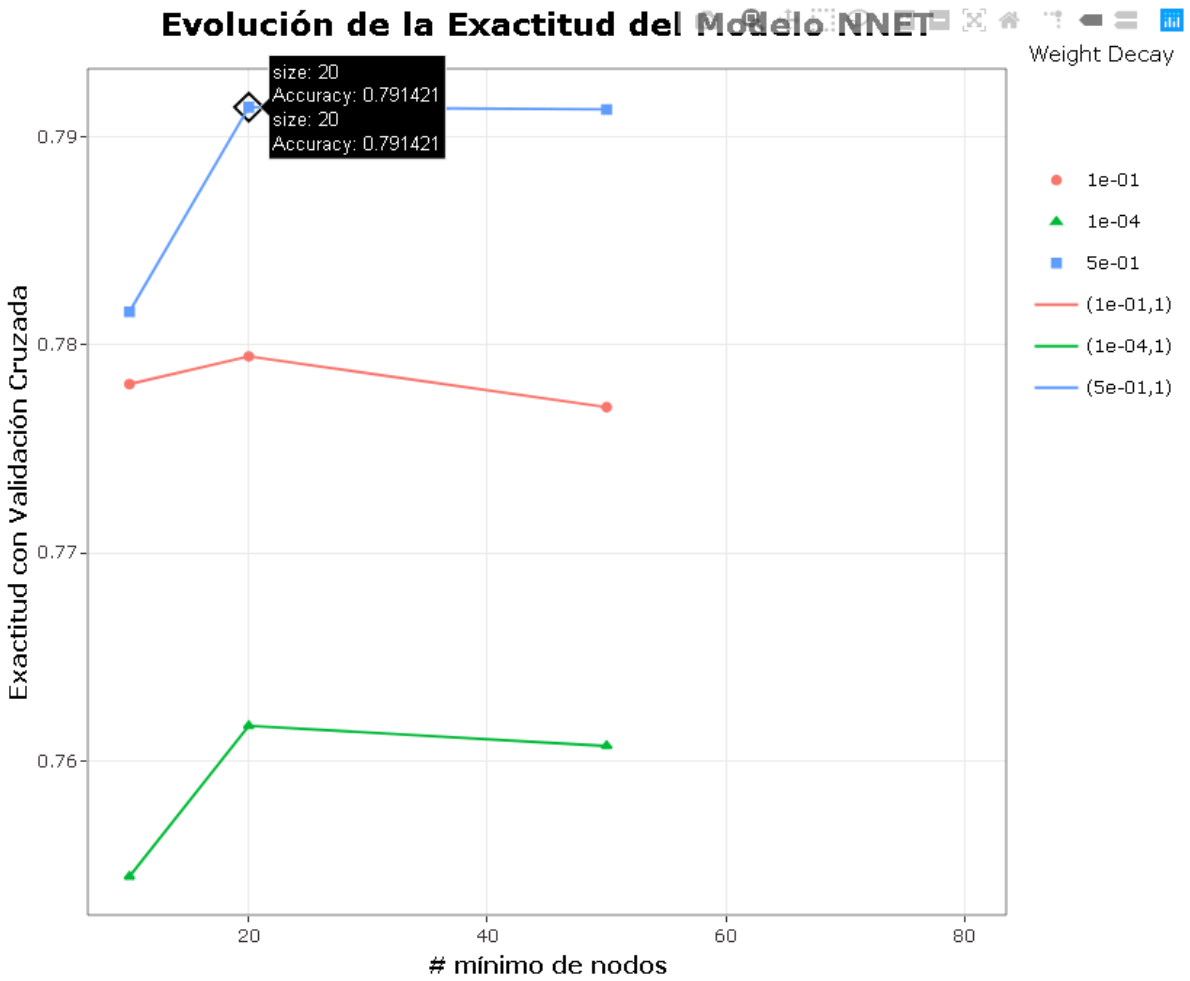


Nota. El eje x toma los valores del hiperparámetro C y el eje y los valores de exactitud que se logran con cada uno de los valores de C. Adicionalmente cada color de la leyenda representa el valor del hiperparámetro sigma. El modelo SVM alcanzó una exactitud de 72.27% con el valor del hiperparámetro sigma igual a 0.001 y el hiperparámetro C igual a 200. Un porcentaje mayor que el obtenido con el modelo KNN, que puede garantizar mayor exactitud en la predicción.

2.2.6.iv. Modelo NNET. El modelo de Redes Neuronales del *Caret* emplea la función *nnet ( )* del paquete *nnet* para crear redes neuronales con una capa oculta. Este algoritmo tiene dos hiperparámetros: *size* que define en número de neuronas en la capa oculta, y *decay* que controla la regularización durante el entrenamiento de la red. Además de estos hiperparámetros la función *nnet* tiene muchos otros argumentos que controlan diferentes procesos de aprendizaje de la red [18]. Lo resultados de este modelo se evidencian en la **Figura 16.**

Figura 16.

Gráfico de Exactitud NNET

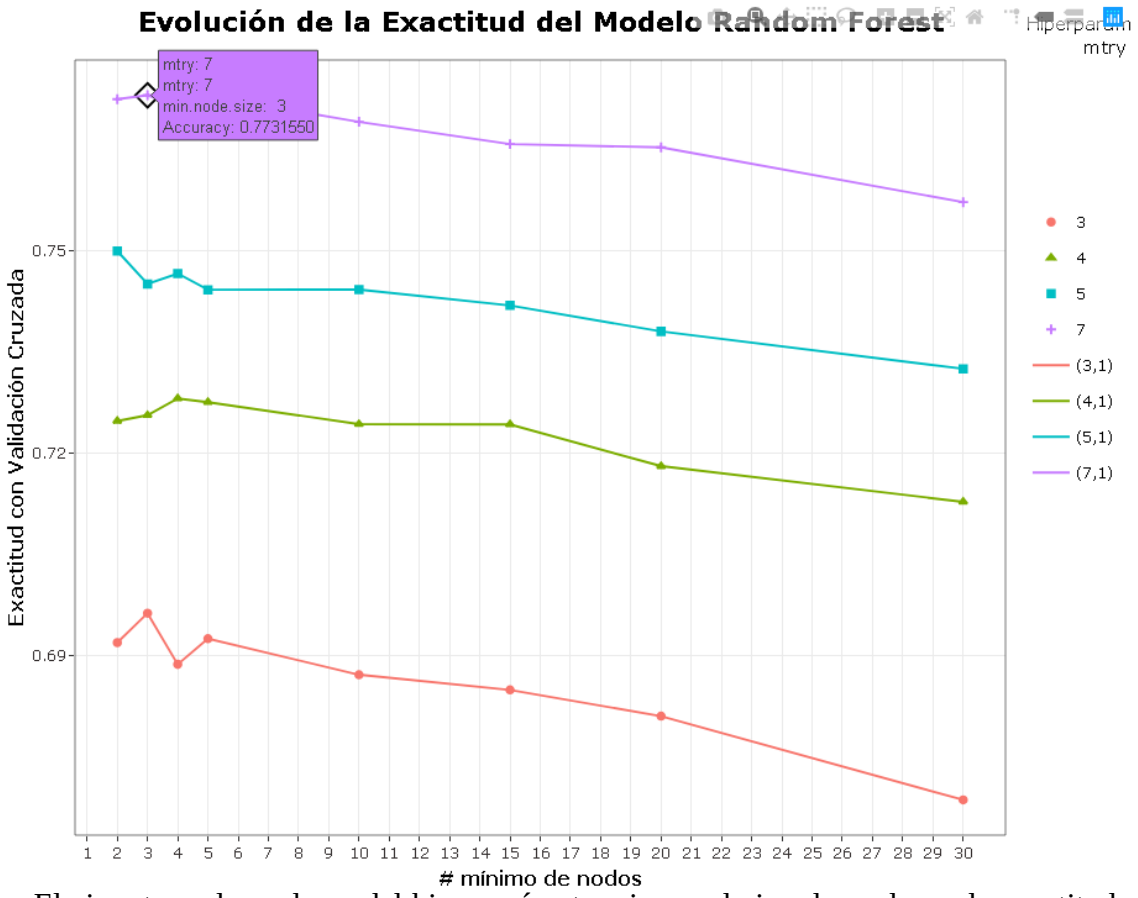


Nota. El eje x toma los valores del hiperparámetro size, y el eje y los valores de exactitud que se logran con cada uno de los valores de size. Adicionalmente cada color de la leyenda representa el valor del hiperparámetro decay. En la gráfica se observa que las rectas solo llegan hasta un valor size = 50, lo que significa que se estableció un rango de valores para "# mínimo de nodos" mayor al límite que permiten los cálculos del modelo. La exactitud de 79.14% se alcanzó con los valores de size=20 y decay = 0.5.

2.2.6.v. Modelo Random Forest. Es un modelo basado en árboles de decisión, los cuales son modelos predictivos formados por reglas binarias (0-1 o si-no) con las que se consigue repartir las observaciones en función de sus atributos y predecir así el valor de la variable respuesta. Los modelos *Random Forest* están formados por un conjunto de árboles de decisión individuales, cada uno entrenado con una muestra ligeramente distinta de los datos de entrenamiento generada mediante remuestreo (*bootstrapping*) [20]. Se programó la **Figura 17.**, la cual permitió observar gráficamente como fue la exactitud de la predicción del modelo al hacer cálculos variando el valor que toman los hiperparámetros *mtry* y *node size*.

Figura 17.

Gráfico de Exactitud RF

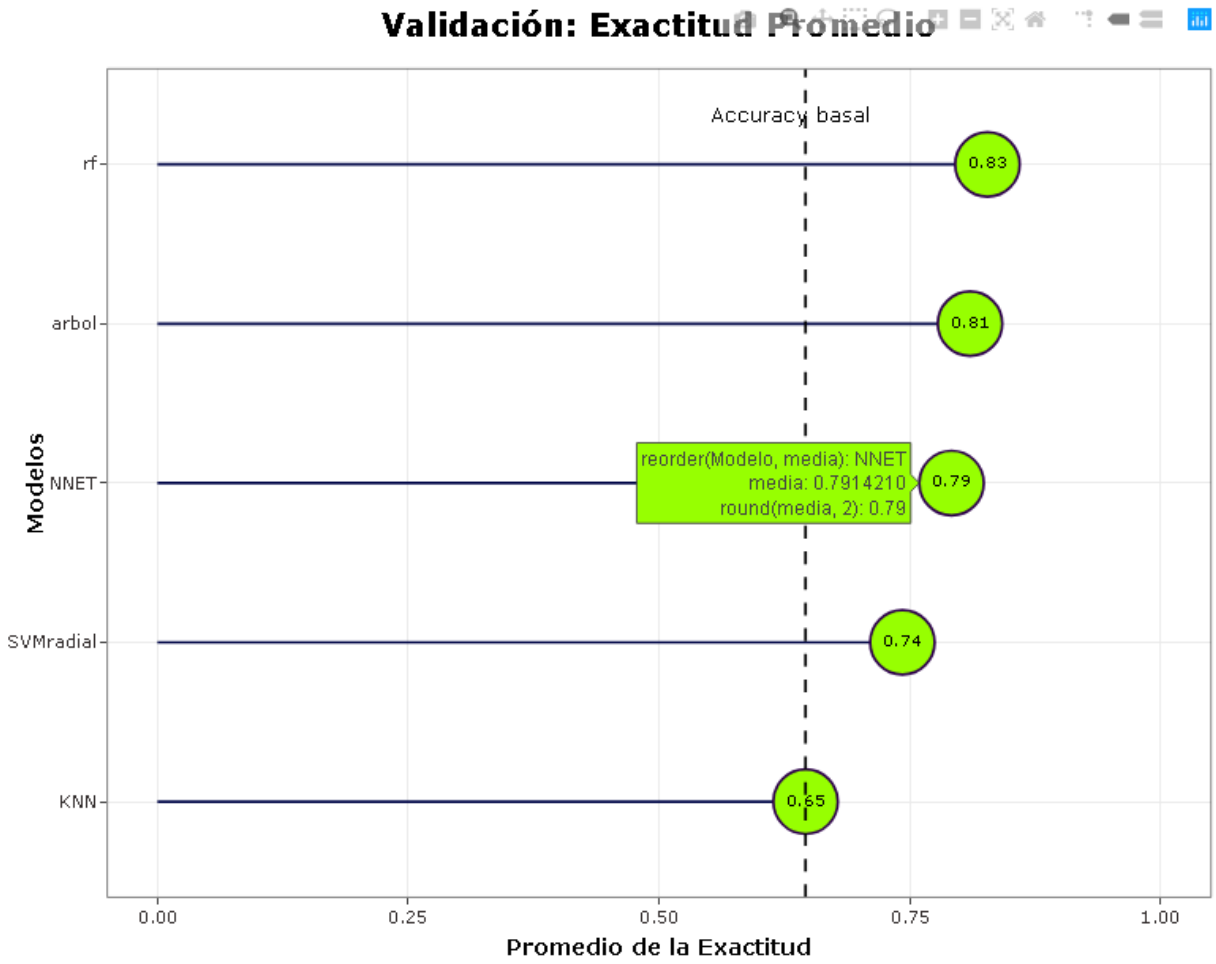


Nota. El eje x toma los valores del hiperparámetro size, y el eje y los valores de exactitud que se logran con cada uno de los valores de size. Adicionalmente cada color de la leyenda representa el valor del hiperparámetro mtry. En la gráfica se observa que los mejores resultados siempre se dieron con un valor de mtry=7, pero la mayor exactitud de 77.31% fue con 3 nodos.

La selección del modelo óptimo se llevó a cabo a partir de la comparación de los valores de exactitud que se obtuvieron según el número de hiperparámetros en cada uno de los modelos, donde los resultados más altos corresponden al “Modelo NNET” y al “Modelo RF”, con 79.14% y 77.31%. Sin embargo, para obtener una respuesta más sólida se programó el gráfico de la **Figura 18.**, en los cuales se realizó un promedio de los valores de exactitud obtenidos de los cálculos de *Resamples* (Remuestreos) que se extrajeron mediante el llamado *$resample* en cada uno de los resultados de cada modelo.

Figura 18.

Comparación de Modelos



Nota. De los cinco modelos programados, el que mejor resultado de exactitud logró al realizar los cálculos fue el modelo Random Forest con un promedio de exactitud de 83%. La línea basal se graficó con el fin de observar que tan exacto es un modelo con respecto al de menor exactitud.

En resumen, el modelo *Random Forest* (“Modelo_RF”) fue seleccionado según los criterios ya mencionados, para realizar la extracción de los resultados asociados a la predicción de los datos de entrenamiento, los datos de test y los datos utilizados para la validación del algoritmo.

3. RESULTADOS

La recolección de resultados de este proyecto inició cuando se finalizó la construcción del dataframe. Debido a que este contiene bastante información, la forma adecuada de mostrar todas las variables que lo conforman, el origen de estas y los valores que toman, fue programando el gráfico de la **Figura 19.** Este es el dataframe que se utilizó en el estudio estadístico, el cual abarca desde el análisis exploratorio hasta la extracción de resultados del modelo *Random Forest*.

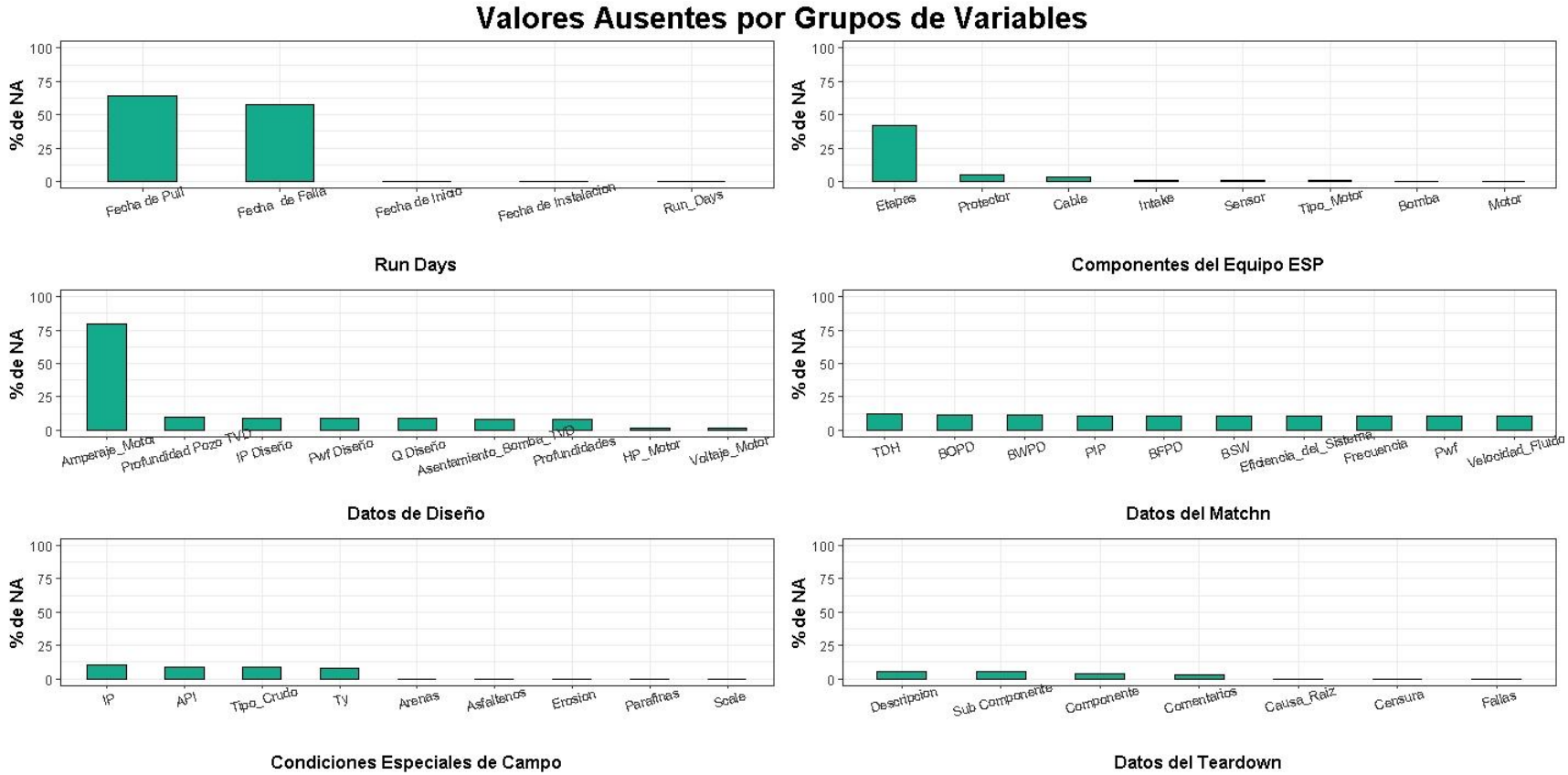
3.1. Resultados del Estudio Estadístico

3.1.1. Resultados del Análisis Exploratorio

Para dar cumplimiento al segundo objetivo, se programaron gráficos para observar los resultados más representativos del análisis exploratorio, los cuales son la base para la interpretación de los resultados finales. Siguiendo el objetivo general y basándose en la información descrita en la metodología, es importante observar cómo es la distribución de los componentes de los equipos ESP en función de la variable respuesta "Causa_Raiz". La **Figura 20.**, muestra el total de "Bombas" que hacen parte del dataframe y el número de veces que se ha instalado cada una, junto con las fallas que ha presentado cada una y las que se encuentran corriendo. Del gráfico se logró extraer cuales son las bombas que más se instalan, cuáles son las que menos se instalan, cuáles son las que más fallan por arena, por asfaltenos o por scale, y cuáles son las bombas en la que más se presentan problemas de fabricación. Para realizar los gráficos de las predicciones del dataframe entrenamiento se extrajo específicamente las cuatro bombas de mayor instalación, cuatro bombas de mediana instalación y cuatro bombas de poca instalación, con el fin de observar los resultados de forma distribuida. De igual forma se extrajo información de la **Figura 21.**, correspondiente a la variable "Motor" y de la **Figura 22.**, correspondiente a la variable "Tipo_Motor". Otro gráfico importante del análisis exploratorio es el Correlograma de la **Figura 23.**, con el que se realizó la selección de 13 variables cuantitativas dentro de un total de 22 variables.

Figura 19.

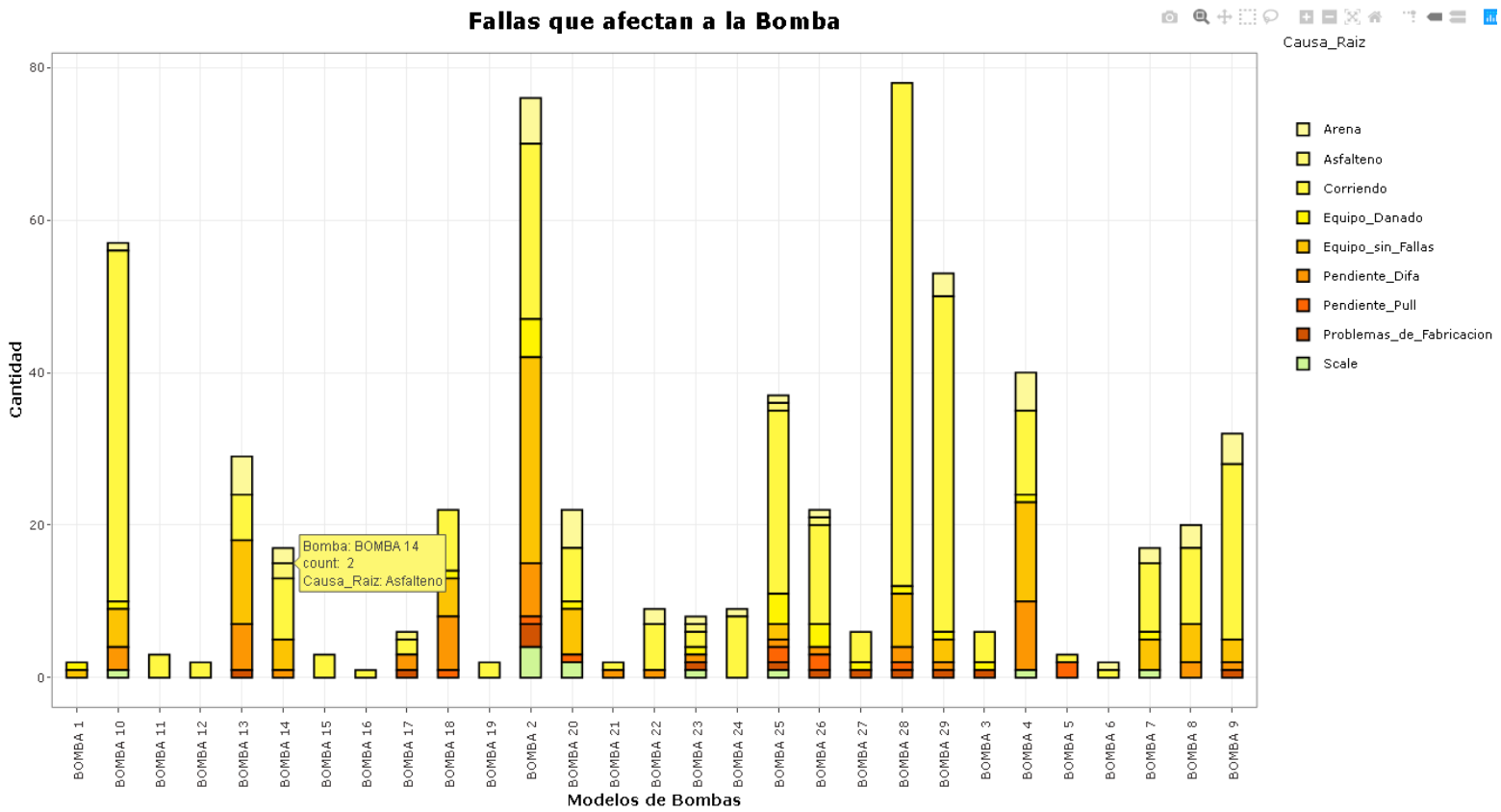
Variables del Dataframe



Nota. 51 variables que conforman el dataframe agrupadas según su origen. Las barras de color representan el porcentaje de valores ausentes por cada una de las variables con relación al total de pozos estudiados.

Figura 20.

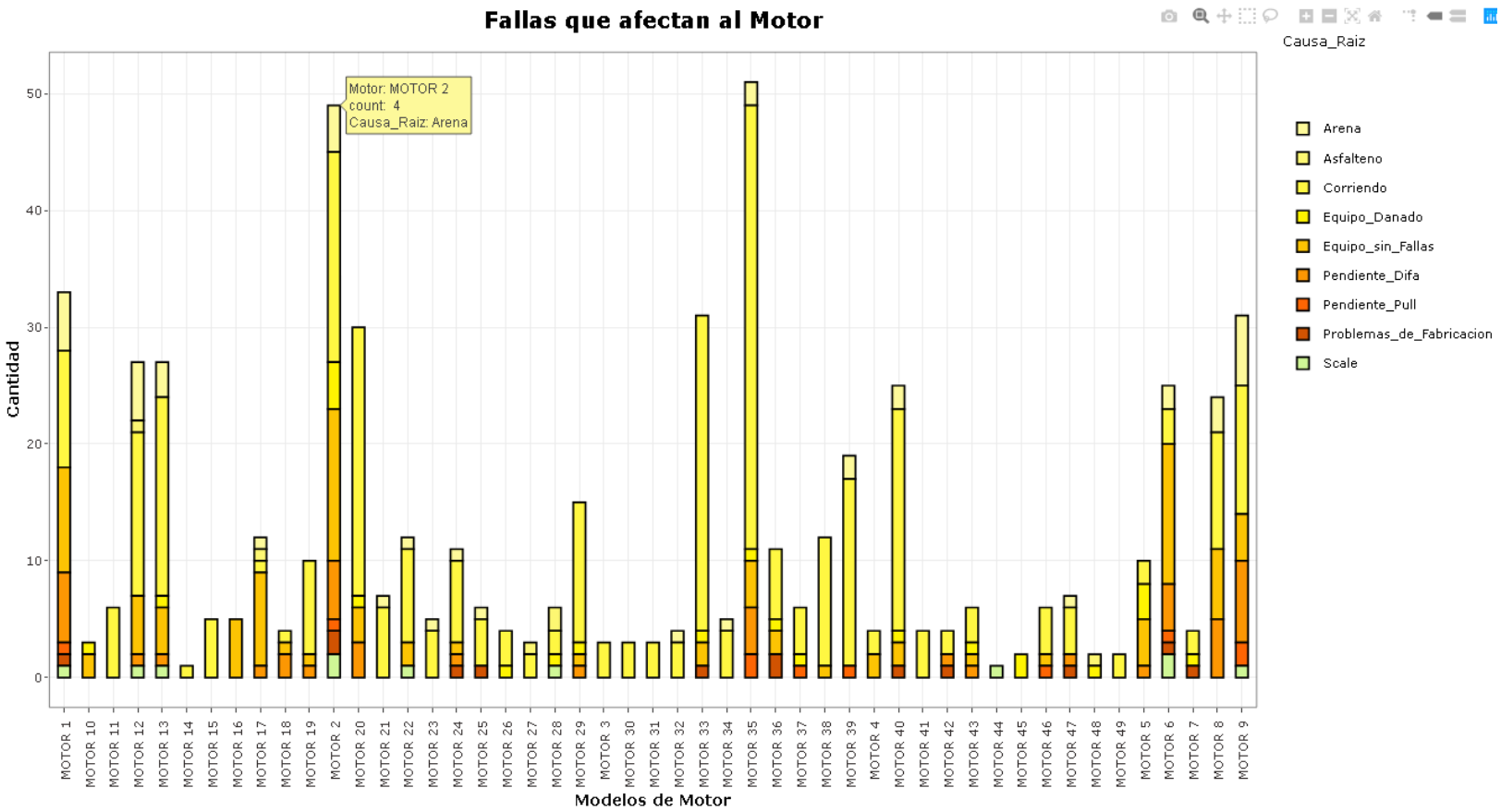
Distribución de la Variable Bomba en Función de la Causa Raíz



Nota. De cada uno de los 29 modelos de bomba diferentes (eje x) que se han instalado en los 586 pozos, se puede extraer información que permite realizar un conteo (eje y) y obtener un porcentaje bomba a bomba, en función de los niveles de la "Causa Raíz" (leyenda), por ejemplo: La Bomba 28 y Bomba 2 se han instalado 78 y 76 veces respectivamente, siendo las más instaladas; La Bomba 13, Bomba 2, Bomba 20 y Bomba 4 son las que más han fallado debido a presencia de Arena; Entre la Bomba 2 y la Bomba 28 se puede decir que la Bomba 28 presenta mejor rendimiento ya que el 86% de las instaladas no han presentado fallas, mientras que de la Bomba 2 solo el 30% de las instaladas no han presentado fallas.

Figura 21.

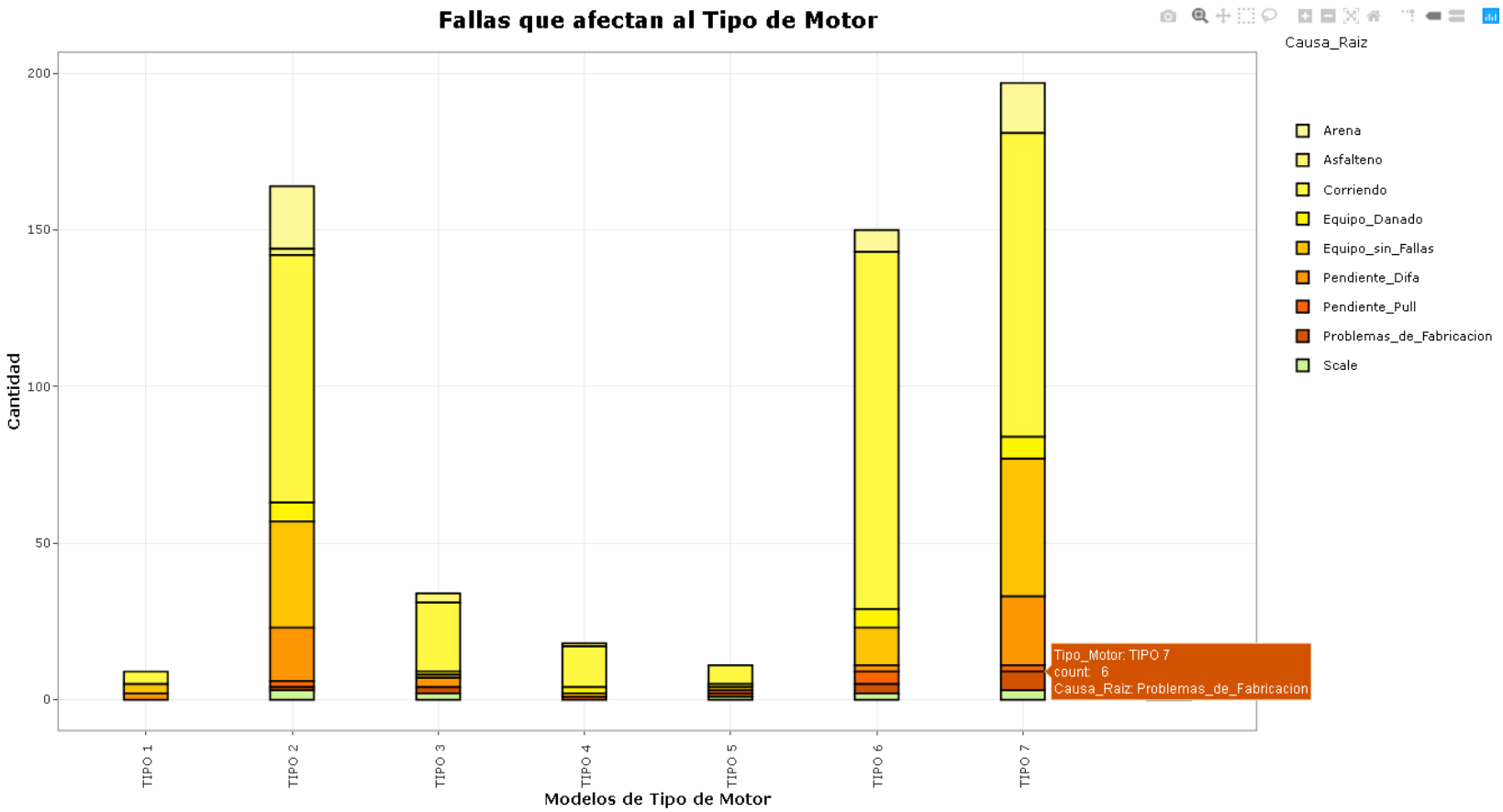
Distribución de la Variable Motor en Función de la Causa Raíz



Nota. De cada uno de los 49 modelos de motor diferentes (eje x) que se han instalado en los 586 pozos, se puede extraer información que permite realizar un conteo (eje y) y obtener un porcentaje motor a motor, en función de los niveles de la "Causa Raíz" (leyenda), por ejemplo: El Motor 35 y Motor2 se han instalado 51 y 49 veces respectivamente, siendo los más instalados; El Motor 12 ha fallado en un 18.5% por Arena, en un 3.7% por asfaltenos y en un 3.7% por scale; El Motor 44 solo presenta una instalación en la que presentó fallas por scale.

Figura 22.

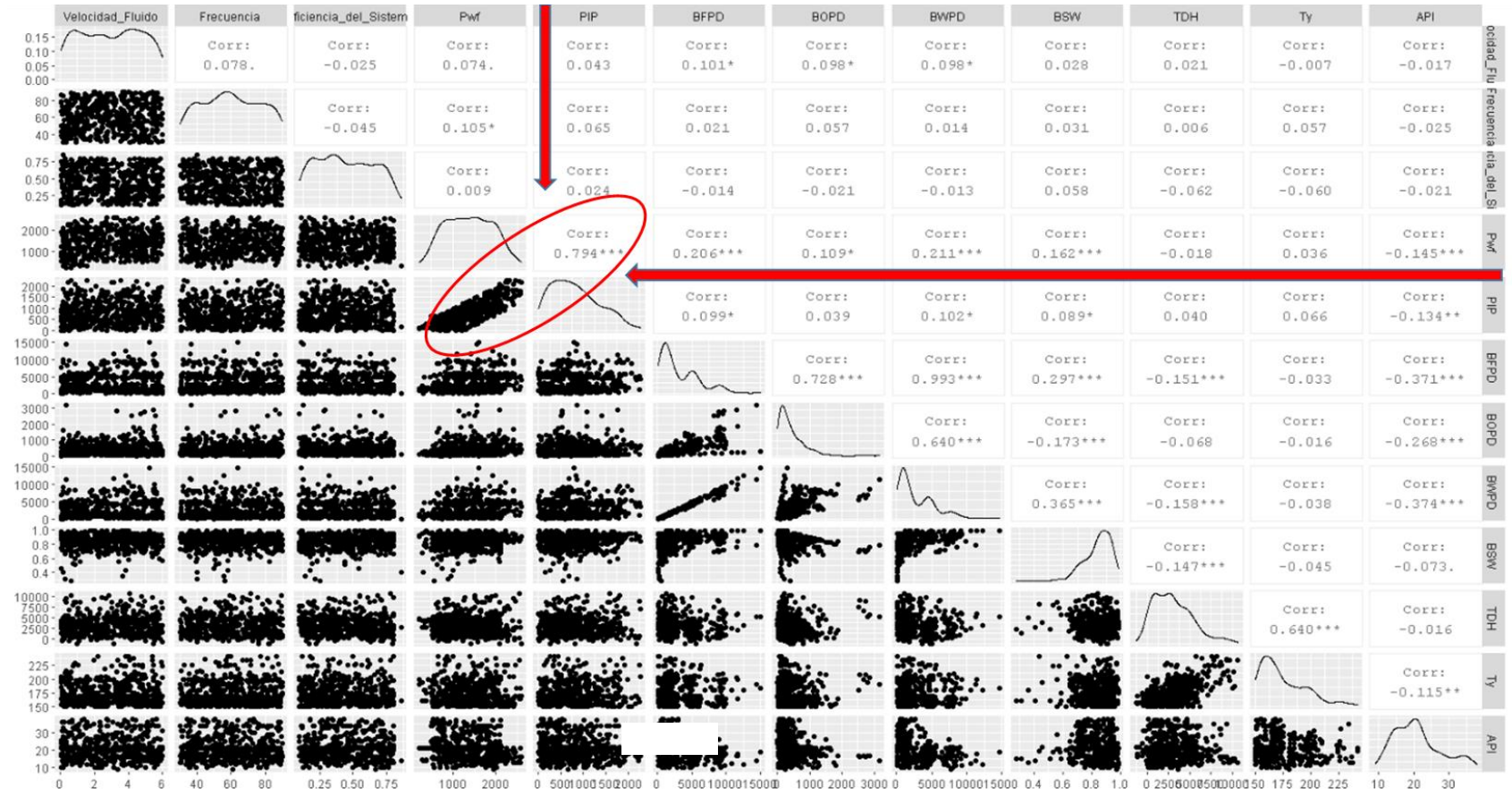
Distribución de la Variable Tipo de Motor en Función de la Causa Raíz



Nota. De cada uno de los 7 modelos de tipo de motor diferentes (eje x) que se han instalado en los 586 pozos, se puede extraer información que permite realizar un conteo (eje y) y obtener un porcentaje modelo a modelo, en función de los niveles de la "Causa Raíz" (leyenda), por ejemplo: Todos los tipos de motor que se han instalado presentaron al menos una falla, a excepción del Tipo 1 que presenta el menor número de instalaciones. El modelo Tipo 6 es el que mejor comportamiento ha tenido ya que de 150 que se han instalado el 76% se encuentra corriendo óptimamente.

Figura 23.

Correlograma



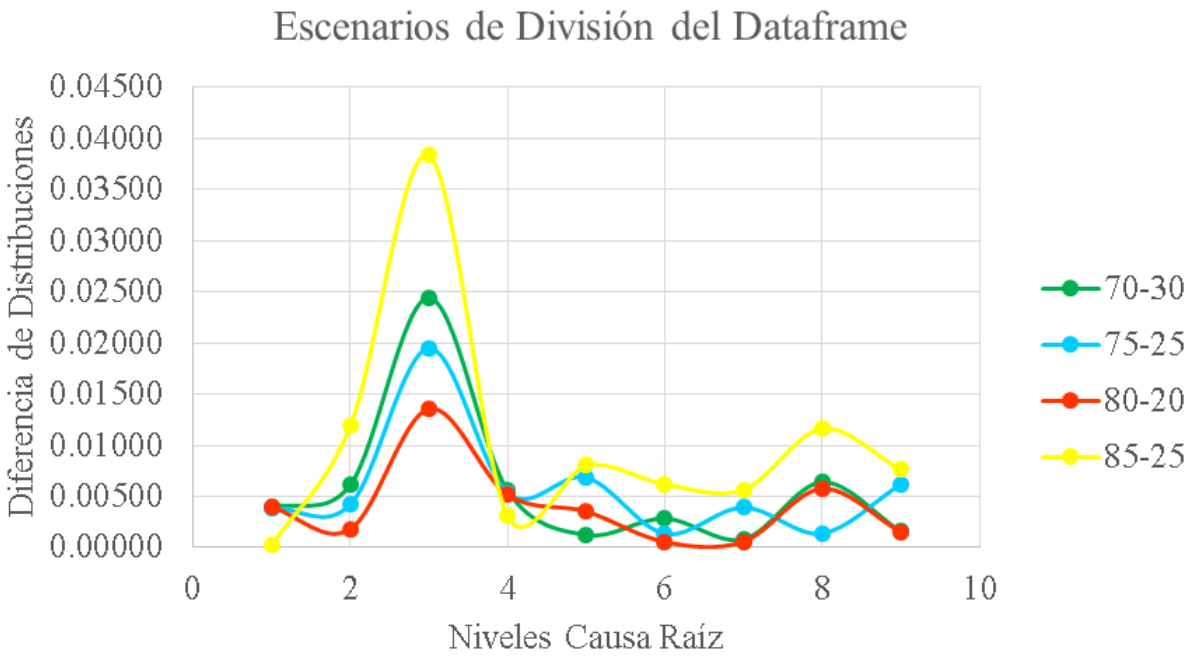
Nota. Gráfico que se programó con las variables cuantitativas del grupo "Datos del Match" y dos variables del grupo "Condiciones Especiales de Campo" (Ty y API). Se utilizó para eliminar las variables cuantitativas que aportan información redundante o poca información al modelo predictivo, dejando solo las más relevantes. La selección se llevó a cabo analizando el valor de "Corr" que se observa entre cada cruce de variables, siguiendo la mostrada en el gráfico; si el valor de Corr >= 0.5 entre dos variables, solo se acepta una de estas; en caso de ser Corr < 0.5, se aceptan ambas variables.

3.1.2. Resultados de los Escenarios de División del Dataframe

La **Figura 24.**, muestra la diferencia de cada uno de los escenarios que se lograron a través de la programación de una tabla estadística que muestra cómo queda la distribución de cada nivel de la variable "Causa_Raíz" en cada uno de los dataframes (entrenamiento y test). Teniendo en cuenta que lo ideal es que la distribución sea lo más equitativa posible, se generaron varios escenarios y en cada uno se calculó la "diferencia" entre las distribuciones del conjunto entrenamiento y el conjunto test; el mejor resultado se obtiene cuando el valor de la "diferencia" se aproxima a cero.

Figura 24.

Escenarios de División



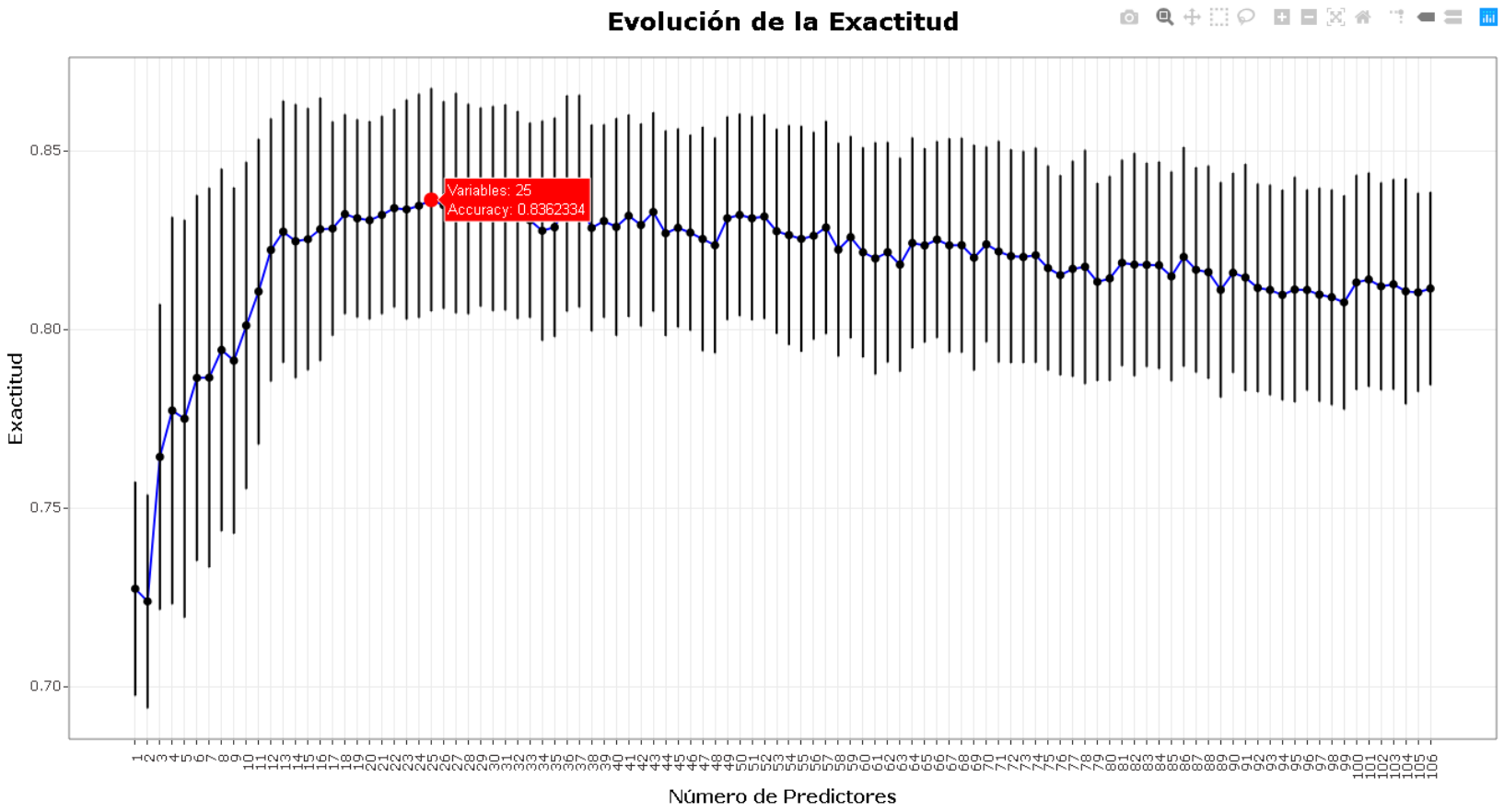
Nota. En el eje x de la gráfica se representan los nueve niveles de la variable "Causa_Raíz" (Nivel 1: Arena; Nivel 2: Asfalteno; Nivel 3: Corriendo; Nivel 4: Equipo Dañado; Nivel 5: Equipo sin Fallas; Nivel 6: Pendiente Difa; Nivel 7: Pendiente Pull; Nivel 8: Problemas de Fabricación; y Nivel 9: Scale). La línea de color rojo representa los valores más aproximados a cero en la mayoría de los niveles (a excepción del nivel 5), por lo cual se seleccionó la división 80% entrenamiento y 20% test.

3.1.3. Resultados de la Selección de Predictores

Seleccionar los predictores óptimos fue el paso más importante de la metodología *Machine Learning* con *Caret*. La selección del número de predictores se realizó a partir de la interpretación de la **Figura 25.**

Figura 25.

Exactitud de la Eliminación Recursiva de Variables



Nota. El punto rojo sobre la gráfica muestra que con 25 predictores se logra el valor más alto de exactitud igual a 83.62%. Este gráfico se programó con el "Dataframe_Entrenamiento_Preprocesado", el cual está conformado por 25 predictores y una variable respuesta, es decir que el resultado óptimo se logra con todas las variables incluidas en este dataframe. En el eje x se puede observar que hay un número mayor de predictores que el que había inicialmente en el dataframe. Esto se debe a que durante el preprocesado se crean nuevas variables a partir de las variables cualitativas como Bomba, Motor y Tipo de Motor.

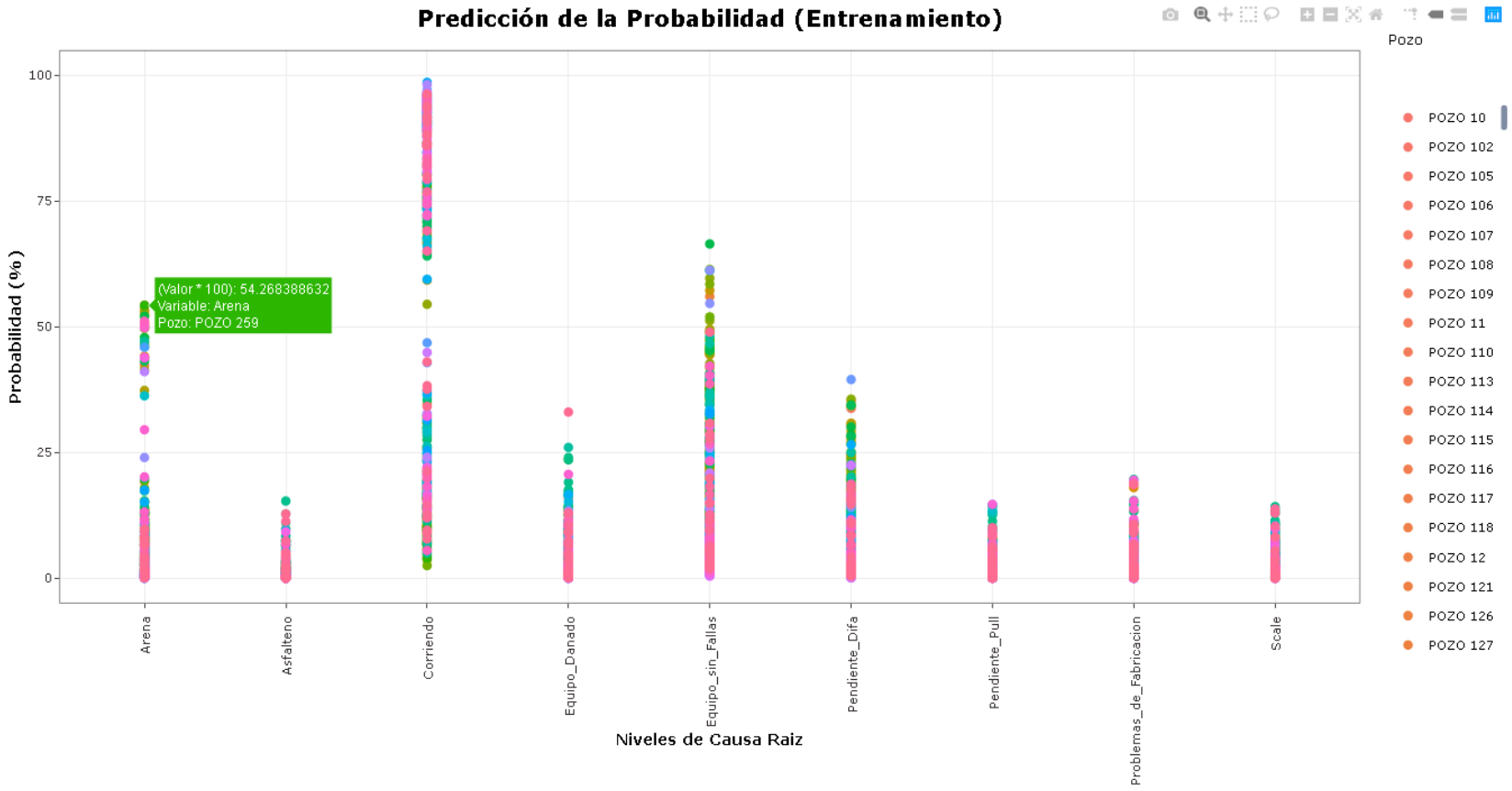
Una vez se estableció que con 25 predictores se logra la mayor exactitud, se recurrió a los métodos de selección de predictores mencionados en la sección de metodología y datos (**Figura 12.**), y se observó que ninguno de los métodos tuvo como resultado un total de 25 predictores (20 en el método de eliminación recursiva de variables y 43 en el método de filtrado), razón por la cual se justificó dejar los mismos 25 predictores de entrada. Se construyeron dos *dataframe* (Dataframe_Entrenamiento_Modelo y Dataframe_Test_Modelo) con el fin de almacenar y reconocer fácilmente los *dataframe* que se utilizaron para el entrenamiento y la predicción. El primero está conformado por 419 pozos y por las siguientes 26 variables: Causa Raíz, Run Days, Bomba, Etapas, Motor, Tipo Motor, HP Motor, Asentamiento Bomba TVD, Velocidad Fluido, Frecuencia, Eficiencia del Sistema, PIP, BFPD, BSW, TDH, Ty, IP, API, Tipo Crudo, Profundidades_Profundo, Profundidades_Somero, Arenas_Si, Asfaltenos_Si, Erosión_Si, Scale_Si, Fallas_Si. Adicionalmente se construyeron otros dos *dataframe* (igualmente con 419 pozos), en los cuales se le agregaron cuidadosamente las siguientes variables sin haberlas sometido al preprocesado de datos: Campo, Pozo, Arenas, Asfaltenos, Erosión y Scale. Estos últimos solo se utilizaron para realizar gráficos de resultados.

3.2. Resultados del Entrenamiento del Modelo

Utilizando el 80% de los datos del "Dataframe_Principal" y el modelo *Random Forest*, se ejecutó el entrenamiento del modelo. Esta ejecución permitió que el modelo aprendiera a identificar predicciones de la variable "Causa_Raíz" con la información de entrada de las 25 variables, con el objetivo de que cuando se le ingrese la información correspondiente al 20% (sin ingresar la variable "Causa_Raiz"), este tenga la capacidad de realizar las predicciones en cada uno de los pozos que hacen parte del "Dataframe_Test_Modelo". La **Figura 26.**, muestra las predicciones que el modelo identificó con los datos de entrenamiento. Estos no hacen referencia a una predicción, sino a la información de entrada suministrada en el 80% transformada de tal forma que el modelo lo reconozca como una predicción, para luego poder ser utilizado para predecir resultados con nuevos datos. Para llegar a este gráfico se unificaron el "Dataframe_Entrenamiento_Modelo" con los resultados de la predicción, los cuales fueron extraídos del modelo ejecutando el comando "Modelo_RF$finalModel$predictions".

Figura 26.

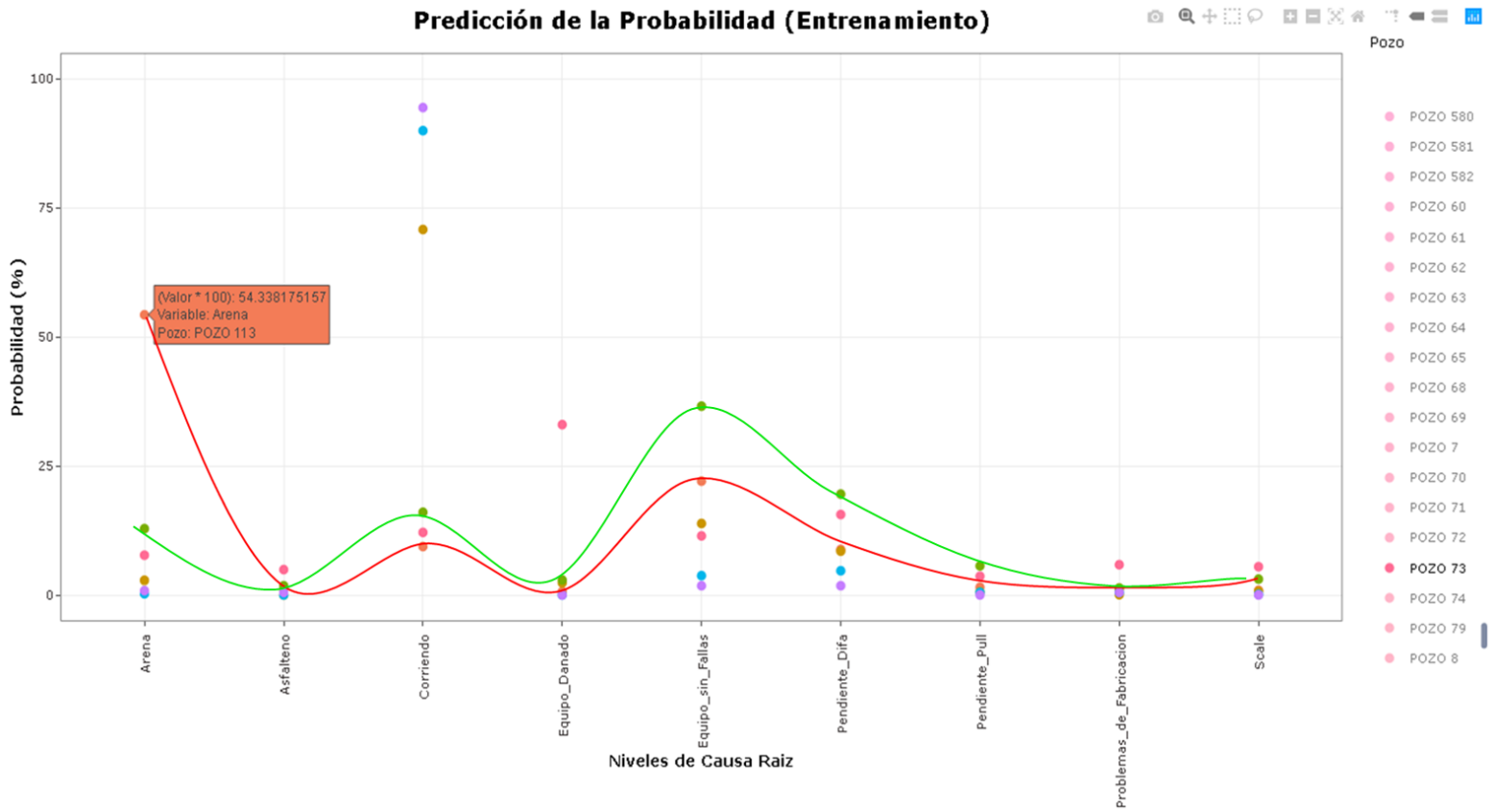
Resultados Generales del Entrenamiento del Modelo



Nota. En el eje x del gráfico se encuentran cada uno de niveles de la variable Causa Raíz con sus respectivos valores de probabilidad de ocurrencia (eje y) en cada uno de los 419 pozos (leyenda: cada color hace referencia a un pozo). El recuadro de color verde muestra que el Pozo 259 tuvo una probabilidad de 54.26%, siendo esta la más alta con respecto a los otros niveles, lo que se traduce en que dicho pozo falló por arena. Este gráfico es un gráfico dinámico que permite que el usuario interactúe para analizar pozo por pozo de forma eficiente, como se observa en la **Figura 27.**

Figura 27.

Análisis Detallado de los Resultados Generales del Entrenamiento del Modelo

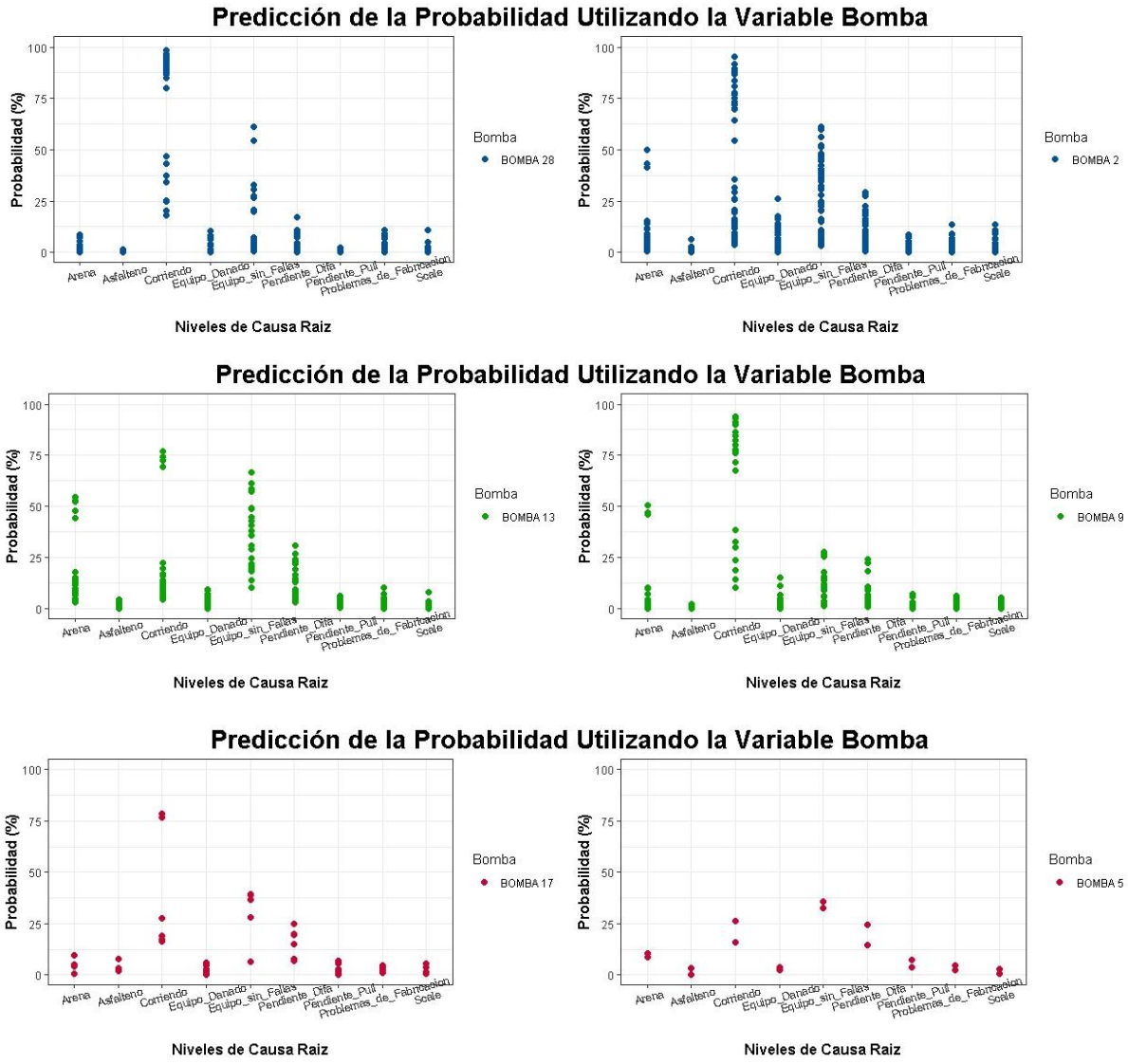


Nota. Manipulando el gráfico anterior se seleccionaron seis pozos de tal forma que se pudiera observar y analizar pozo por pozo. Pozo 113 (Rojo), Pozo 233 (verde), Pozo 73 (rosado), Pozo 170 (amarillo), Pozo 475 (Morado), Pozo 387 (Azul). La línea roja representa la distribución de la probabilidad que tiene el Pozo 113, en donde su valor más alto corresponde al nivel Arena con 54.34%. La línea verde del Pozo 233 indica que el valor más alto de probabilidad corresponde a Equipo sin Fallas. De esta forma se puede analizar cada pozo comprendido en el resultado general del entrenamiento del modelo, para poder conocer a detalle la información que hace parte del 80%.

Además de poder observar los resultados generales, los resultados del entrenamiento del modelo permitieron programar gráficos en los que se observa la distribución de probabilidad de las fallas que han presentado cada uno de los componentes del equipo ESP que se han instalado en cada uno de los 419 pozos. Debido a la gran cantidad de niveles que tienen las variables correspondientes a los componentes de los equipos, solo se realizaron gráficos para las variables Bomba, Motor y Tipo de Motor. Para la variable Bomba (**Figura 28.**) se seleccionaron seis bombas de la siguiente manera: las dos que más se han instalado, dos del medio, y las dos que menos se han instalado. La misma metodología se aplicó para la variable Motor (**Figura 29.**), y de la variable Tipo de Motor (**Figura 30.**) fueron incluidos todos los niveles, ya que solo cuenta con siete.

Figura 28.

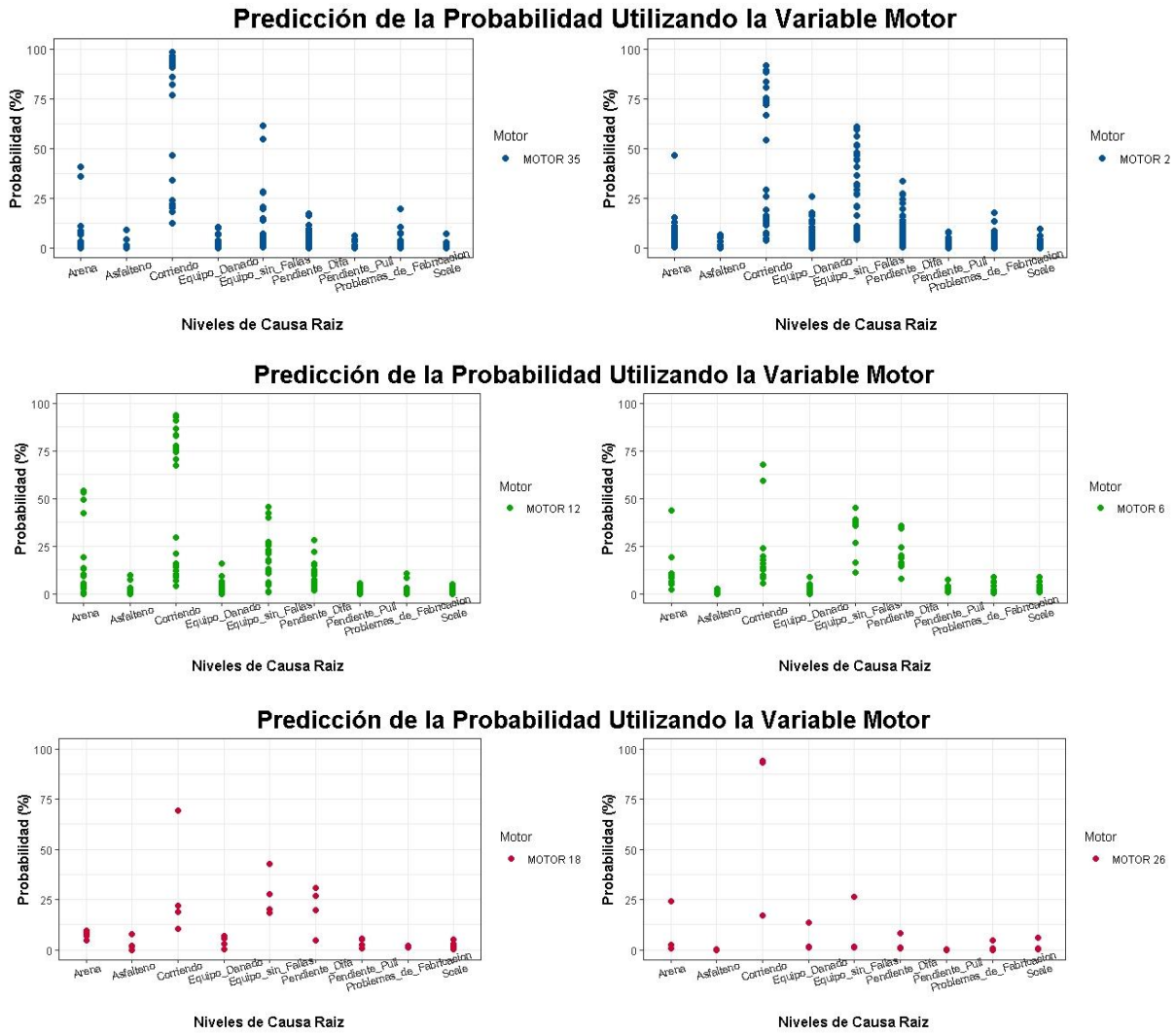
Gráficos de los Resultados por Componente (Bomba)



Nota. En color azul las bombas que más se instalan, en verde las de instalación promedio y en rojo las que menos se instalan. De las anteriores la Bomba 28 es la que más repeticiones tiene en el nivel "Corriendo", lo que significa que es la que mejor resultados ha presentado. En la Bomba 13 se observa varios puntos sobre el 50% del nivel "Arena", lo que demuestra que ha fallado en varias instalaciones cuando del pozo llega arena a la bomba. La bomba 5 se ha instalado en dos ocasiones y se observa que en ambas ocasiones presentaron buen comportamiento ya que en la distribución de la probabilidad no se evidencian puntos de alta probabilidad sobre los niveles arena, Asfalteno y scale.

Figura 29.

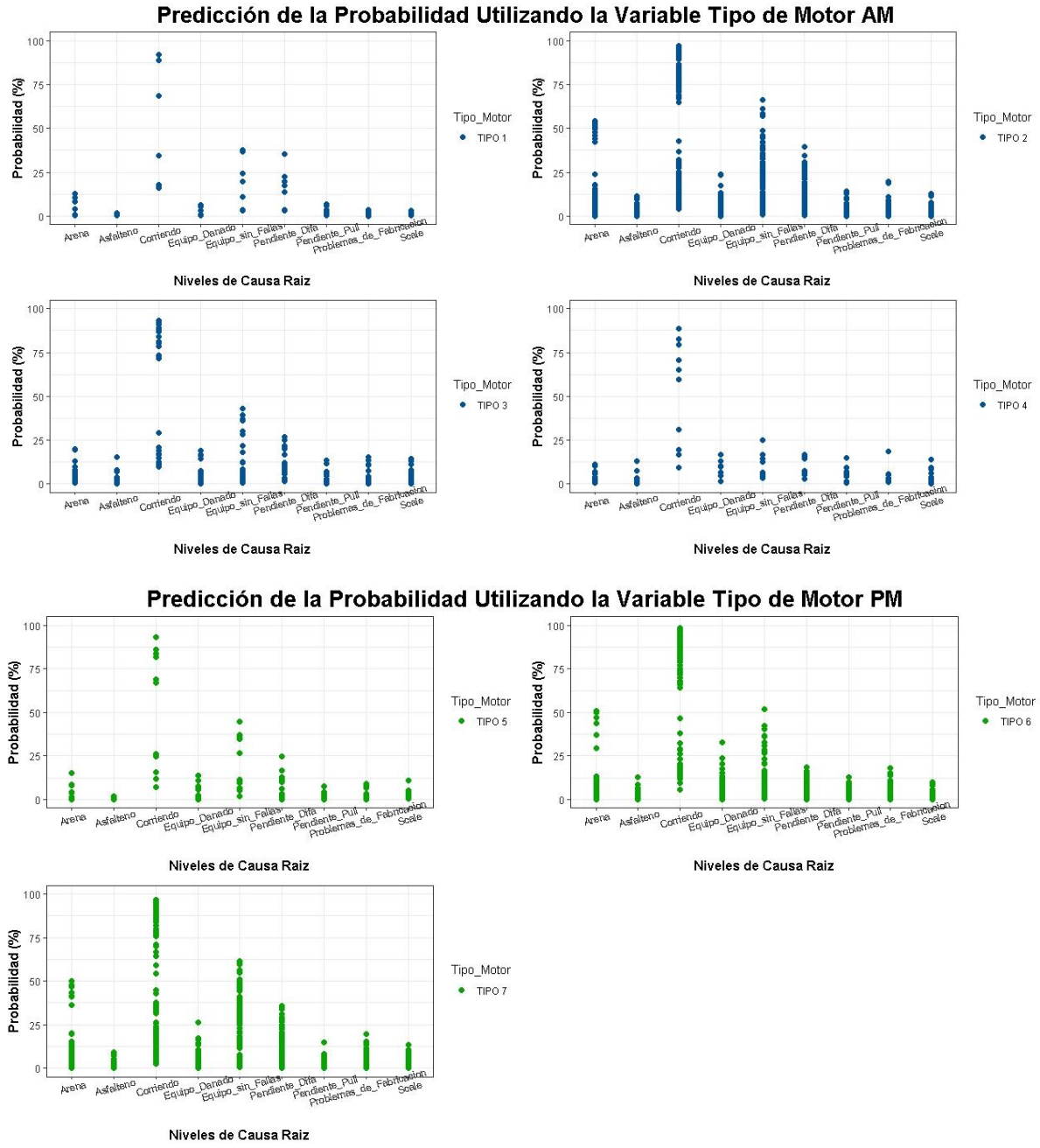
Gráficos de los Resultados por Componente (Motor)



Nota. En color azul los motores que más se instalan, en verde las de instalación promedio y en rojo las que menos se instalan. El Motor 2 presenta el mejor comportamiento en cada pozo que se instaló debido a que la mayoría de sus puntos con alta probabilidad son del nivel “Corriendo” y “Equipo sin Fallas”. El Motor 12 igualmente tiene la mayoría de los puntos de alta probabilidad en el nivel “Corriendo”, pero se destacan tres puntos sobre el 50% del nivel “Arena”. El Motor 26 se ha instalado en dos ocasiones, una de estas se encuentra corriendo óptimamente como se puede evidenciar en la distribución con un porcentaje de probabilidad cercano al 100%.

Figura 30.

Gráficos de los Resultados por Componente (Tipo de Motor)



Nota. En color azul los de motor asincrónico y en verde los motores de imanes permanentes. Comparando entre ambos colores se observa que la mayoría que se instalan son de tipo PM, especialmente el Tipo 6 y el Tipo 7. El Tipo 2, Tipo 6 y Tipo 7 son los que más presentan fallas por presencia de "Arena", lo cual se debe a que son los modelos de tipo de motor que más se instalan.

La información mostrada anteriormente resume parte de la información que contiene el 80% del "Dataframe_Principal" mostrada como resultado de las predicciones extraídas luego del entrenamiento del modelo. Con esta información que almacena el modelo, se logró generar las predicciones que se muestran en los siguientes resultados.

3.3. Resultados de la Predicción del Conjunto Test

El conjunto test está conformado por el 20% de los datos restantes del "Dataframe_Principal", los cuales se almacenaron en el "Dataframe_Test_Modelo" para identificarlo fácilmente al momento de ingresarlo en el código. Este *dataframe* contiene 103 pozos y las mismas variables que el "Dataframe_Entrenamiento_Modelo". En el algoritmo mostrado en la **Figura 13.**, en la parte de predicciones (parte C) se encuentra la línea de código en la cual se ingresó este conjunto de datos como *newdata* (nueva información). En la función *predict ( )* se colocó como argumento el modelo que ya fue entrenado, en este caso el "Modelo_RF". Este se ejecutó de dos formas: la primera con el argumento type = "prob" que dio como resultado la predicción de la variable "Causa_Raíz" con valores numéricos, y la segunda con el argumento type = "raw" con la que se obtuvo el resultado exacto de la predicción en cada pozo de forma cualitativa, como se observa en la **Tabla *6*.** Los cálculos internos que realiza el modelo no tiene en cuenta del *dataframe* la columna "Causa_Raiz", para poder predecirla a partir de la información con la cual se entrenó el modelo; esto fue un factor importante al momento de realizar la validación de la capacidad predictiva del modelo con información que no hizo parte del estudio.

Visualmente los resultados generales de la predicción se observan en la **Figura 31.** A partir de este gráfico se programaron seis gráficos con el objetivo de mostrar la predicción de forma detallada y relacionándola con las variables Bomba (**Figura 32.**), Motor (**Figura 33.**) y Tipo de Motor (**Figura 34.**). Estos gráficos que se muestran a continuación (con los puntos de color verde) hacen referencia a las fallas indirectas, es decir a los niveles de "Causa_Raíz" como Arena, Asfalteno y Scale, los cuales ocurren por causas ajenas a los equipos ESP. Estos son los más importantes a analizar ya que corresponden a respuestas asociadas a las condiciones especiales de campo. También se encuentra Equipo sin Fallas, en el cual el cliente decide parar la operación, siendo ajeno a los equipos ESP de la Compañía A en ambos casos.

Tabla 6.

Resultados Generales de la Predicción del Conjunto Test

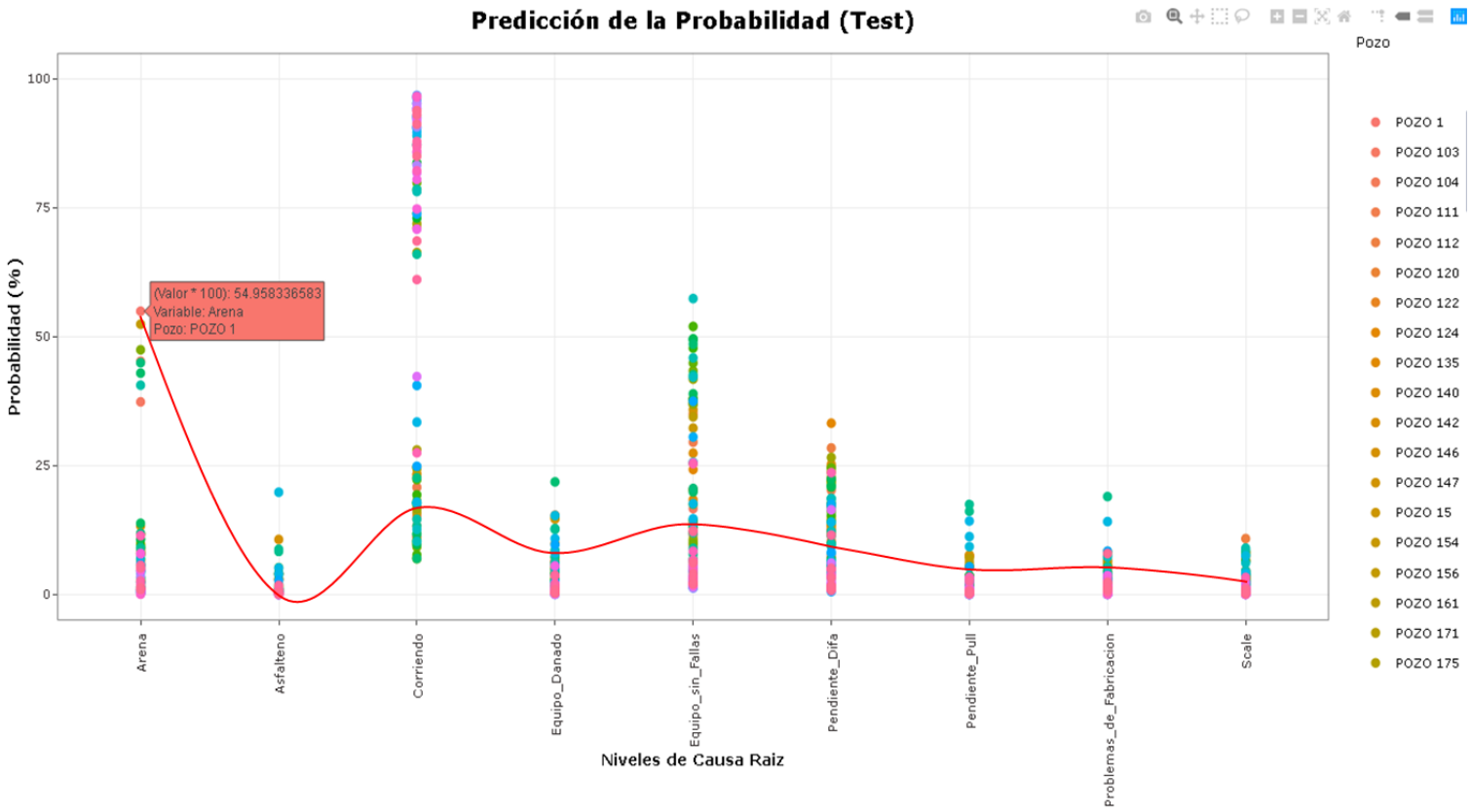
Resultados de la Predicción con el Conjunto Test

Pozo	Valores Originales Causa raiz	Predicciones Causa raiz	Arena	Asfalteno	Corriendo	Equipo_Danado	Equipo_sin_Fallas	Pendiente_Difa	Pendiente_Pull	Problemas_de_Fabricacion	Scale
POZO 1	Arena	Arena	0.549583	0.007658	0.112356	0.064641	0.094581	0.074091	0.033118	0.045669	0.018303
POZO 6	Corriendo	Corriendo	0.012322	0.000846	0.869705	0.013206	0.065459	0.017098	0.004136	0.004105	0.013123
POZO 15	Corriendo	Corriendo	0.014089	0.002266	0.872753	0.009628	0.052532	0.035828	0.001407	0.007969	0.003529
POZO 19	Corriendo	Corriendo	0.008271	0.002746	0.785575	0.019578	0.107663	0.050330	0.010402	0.008025	0.007410
POZO 24	Corriendo	Corriendo	0.011432	0.006118	0.837330	0.012003	0.067871	0.044334	0.005077	0.006849	0.008987
POZO 32	Corriendo	Corriendo	0.022980	0.011249	0.786251	0.021190	0.063007	0.037988	0.031490	0.007040	0.018805
POZO 37	Equipo_sin_Fallas	Corriendo	0.040574	0.038890	0.334046	0.108286	0.146394	0.139997	0.141967	0.009724	0.040122
POZO 41	Corriendo	Corriendo	0.022892	0.027118	0.738293	0.023450	0.046202	0.060104	0.053516	0.012036	0.016388
POZO 45	Corriendo	Corriendo	0.008130	0.000404	0.921135	0.009925	0.030754	0.018745	0.001223	0.004612	0.005072
POZO 48	Corriendo	Corriendo	0.035152	0.003434	0.833101	0.012862	0.051244	0.038738	0.007833	0.003301	0.014334
POZO 59	Problemas_de_Fabricacion	Corriendo	0.113457	0.008025	0.274576	0.036751	0.253301	0.236013	0.032637	0.012822	0.032419
POZO 67	Corriendo	Corriendo	0.026038	0.005662	0.878908	0.018841	0.030971	0.007454	0.007123	0.021328	0.003676
POZO 76	Corriendo	Corriendo	0.056076	0.016679	0.610913	0.036351	0.123592	0.114919	0.026098	0.005619	0.009752
***13 renglones de 103 (omitidos 90 renglones)*											

Nota. En la columna de color verde se observa el nivel que tomaba cada pozo cuando se construyó el dataframe. En la columna de color rojo se observa el resultado que arrojó la predicción (raw), que también se puede interpretar al observar los valores numéricos de la predicción (prob) de cada renglón donde el valor más alto hace referencia al resultado de la predicción. En el Pozo 37 y Pozo 59 se observa que la predicción es diferente al valor original, lo que demuestra que el modelo si tiene la capacidad de predecir nuevos resultados a partir de la experiencia acumulada durante el entrenamiento.

Figura 31.

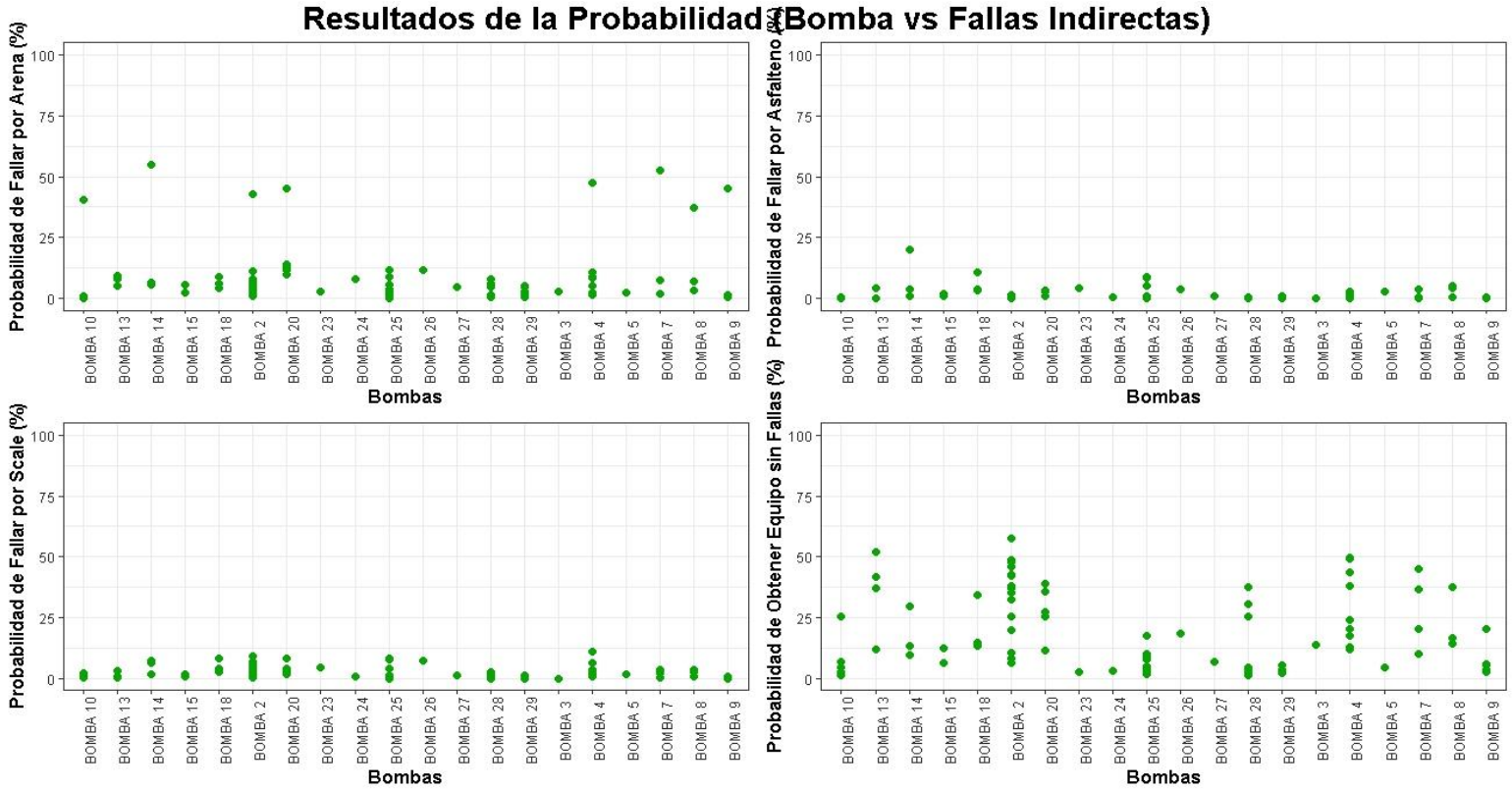
Resultados Generales de la Predicción del Conjunto Test



Nota. En el eje x del gráfico se encuentran cada uno de niveles de la variable Causa Raíz con sus respectivos valores de probabilidad de ocurrencia (eje y) en cada uno de los 103 pozos (leyenda: cada color hace referencia a un pozo). La línea roja se trazó para demostrar que el valor de la probabilidad más alto que obtuvo el Pozo 1 luego de la predicción fue el del nivel "Arena". Como esto fue una predicción, esto se interpretó de la siguiente manera: si se requiere instalar un equipo ESP (con modelo de Bomba 14, modelo de Motor 24 y modelo de tipo de motor Tipo 7) en el Pozo 1, se tiene una probabilidad del 54.95% de que este falle por causa de la "Arena" presente en el pozo. De la misma forma utilizando este gráfico se puede analizar cada uno de los 102 pozos restantes.

Figura 32.

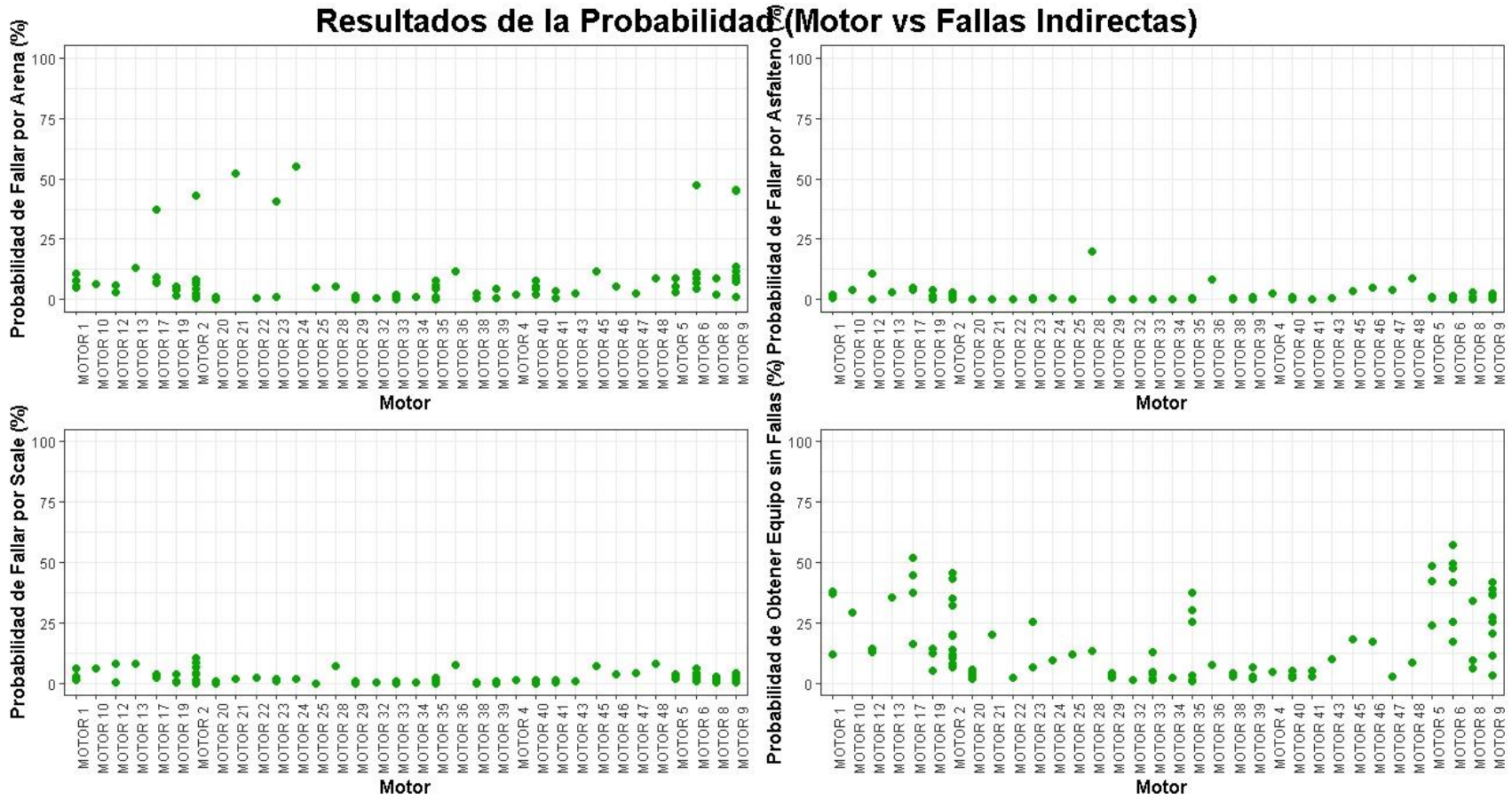
Resultados de la Predicción por Componente (Bomba)



Nota. Se observan cuatro gráficos asociados a la probabilidad de fallas indirectas que pueden llegar a presentar las bombas: Arena (esquina superior izquierda), Asfalteno (esquina superior derecha), Scale (esquina inferior izquierda) y Equipo sin Fallas (esquina inferior derecha). La interpretación de este gráfico se hizo de la siguiente manera: se observaron los puntos de mayor probabilidad y se determinó que una probabilidad por debajo al 25% no influye en resultado de la predicción, por lo cual la probabilidad de que la bomba falle por Scale o por Asfaltenos es muy baja. Por el contrario, como sucede en el cuadro de "Arena" donde se observan ocho puntos que están en el rango entre 40% y 60% de probabilidad, lo que si indica que esos modelos de bomba tienen una probabilidad promedio del 50% de fallar a causa de Arena.

Figura 33.

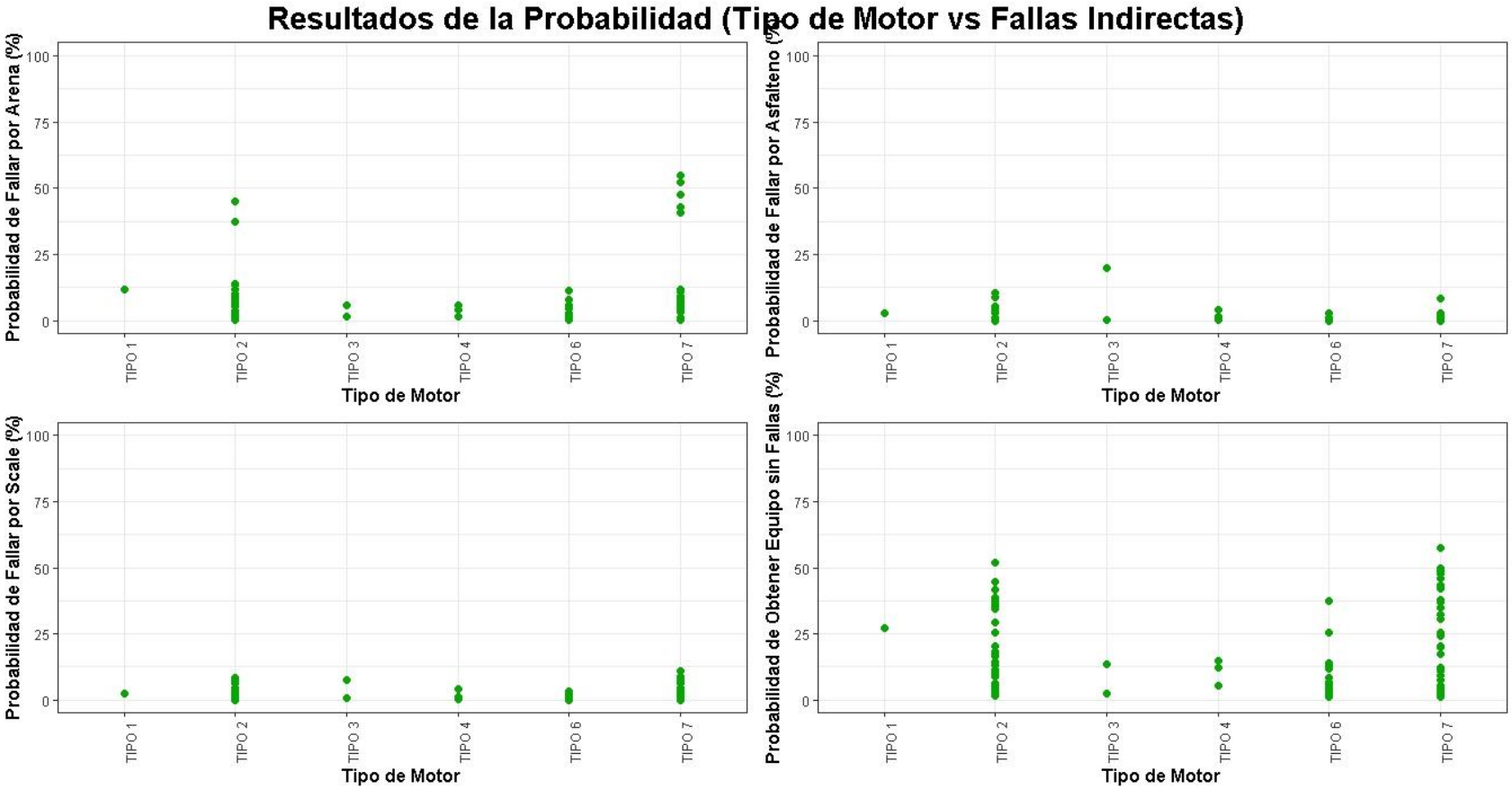
Resultados de la Predicción por Componente (Motor)



Nota. Siguiendo el mismo procedimiento de análisis de la gráfica anterior, la probabilidad de que un Motor falle a causa de Scale o Asfalteno es muy baja, sim embargo el Motor 28 presenta una probabilidad cercana al 25% de fallar por asfalteno. También se determinó que siete motores pueden llegar a presentar fallas por arena debido a la alta probabilidad que arrojó la predicción.

Figura 34.

Resultados de la Predicción por Componente (Tipo de Motor)



Nota. Con este gráfico se confirma que la predicción arrojó resultados que, en cuanto a las condiciones especiales de campo, la que tiene mayor probabilidad de causar fallas en los componentes de los equipos ESP es la “Arena”.

En las gráficas anteriores no se analizaron los demás niveles de la variable "Causa_Raíz" debido a que no entraban dentro de las condiciones especiales de campo, lo que generó que perdieran importancia al momento de analizar los resultados. Sin embargo, fue necesario incluirlas a lo largo del estudio para poder tener la suficiente información, la cual permitió realizar las predicciones bajo la menor influencia por parte de la información de entrada.

3.4. Validación de Resultados

Cada vez que se desee predecir nuevos datos estos deben ser sometidos a este algoritmo desde el paso de preprocesado de datos, hasta ser pasados como *newdata* en la etapa de predicción del modelo *Random Forest*. Esto puede ser un proceso largo y complejo, razón por la cual se creó un archivo en Excel con dos ventanas. Una de estas es una tabla para ingresar cada uno de los valores de los pozos nuevos que se deseen predecir, y la otra contiene un *dataframe* programado de tal forma que cada valor que se ingresa en la otra ventana se almacene en este en su respectiva columna, pero con los valores numéricos centrados y normalizados, de tal forma que el modelo predictivo los acepte sin necesidad de someter los nuevos datos al algoritmo de preprocesado.

Por temas de confidencialidad, la validación no se realizó con datos reales de pozos que no hacían parte de este proyecto. Para evitar la estimación de valores, se seleccionaron un total de 11 pozos al azar del "Dataframe_Principal". Los valores de cada pozo fueron ingresados en la tabla que se observa en la **Figura 35.**, luego mediante la programación en Excel fueron almacenados y transformados en un *dataframe*, para finalmente poderlos cargar en R, donde fueron almacenados como "Dataframe_Validación". El siguiente paso fue someter este *dataframe* al algoritmo de predicción de la misma forma que se realizó en la predicción del conjunto test. Finalmente se extrajeron los resultados de las predicciones y se graficaron de igual forma: en un gráfico general de resultados, como se observa en la **Figura 36.** Adicionalmente, como se observa en la **Figura *38*.**, se calculó el porcentaje de error entre los valores reales (resultados de predicciones con los 419 pozos del 80%) y la predicción que se obtuvo como resultado en la validación de cada uno de los tres pozos estudiados en la **Figura 37.**

Figura 35.

Ingreso de Datos Nuevos

Campo	Pozo	Arenas (Si - No)	Asfaltenos (Si - No)	Erosion (Si - No)	scale (Si - No)	Fallas (Colocar todos Si)	Causa Raiz (Dejar en blanco)	Run Days	Bomba	Etapas
CAMPO 40	POZO 405	No	No	No	No	Si		74	BOMBA 29	85
CAMPO 18	POZO 160	Si	No	No	No	Si		589	BOMBA 13	150
CAMPO 40	POZO 463	No	No	No	No	Si		2076	BOMBA 10	67
CAMPO 40	POZO 388	No	No	No	No	Si		2075	BOMBA 10	100
CAMPO 18	POZO 162	No	No	No	No	Si		374	BOMBA 25	171
CAMPO 19	POZO 262	Si	No	No	No	Si		156	BOMBA 20	32
CAMPO 13	POZO 74	No	Si	No	No	Si		27	BOMBA 1	30
CAMPO 19	POZO 263	No	Si	No	No	Si		1065	BOMBA 20	100
CAMPO 24	POZO 308	No	No	No	Si	Si		135	BOMBA 23	80
CAMPO 38	POZO 324	No	No	No	Si	Si		197	BOMBA 2	70
CAMPO 3	POZO 109	No	No	Si	Si	Si		97	BOMBA 20	100

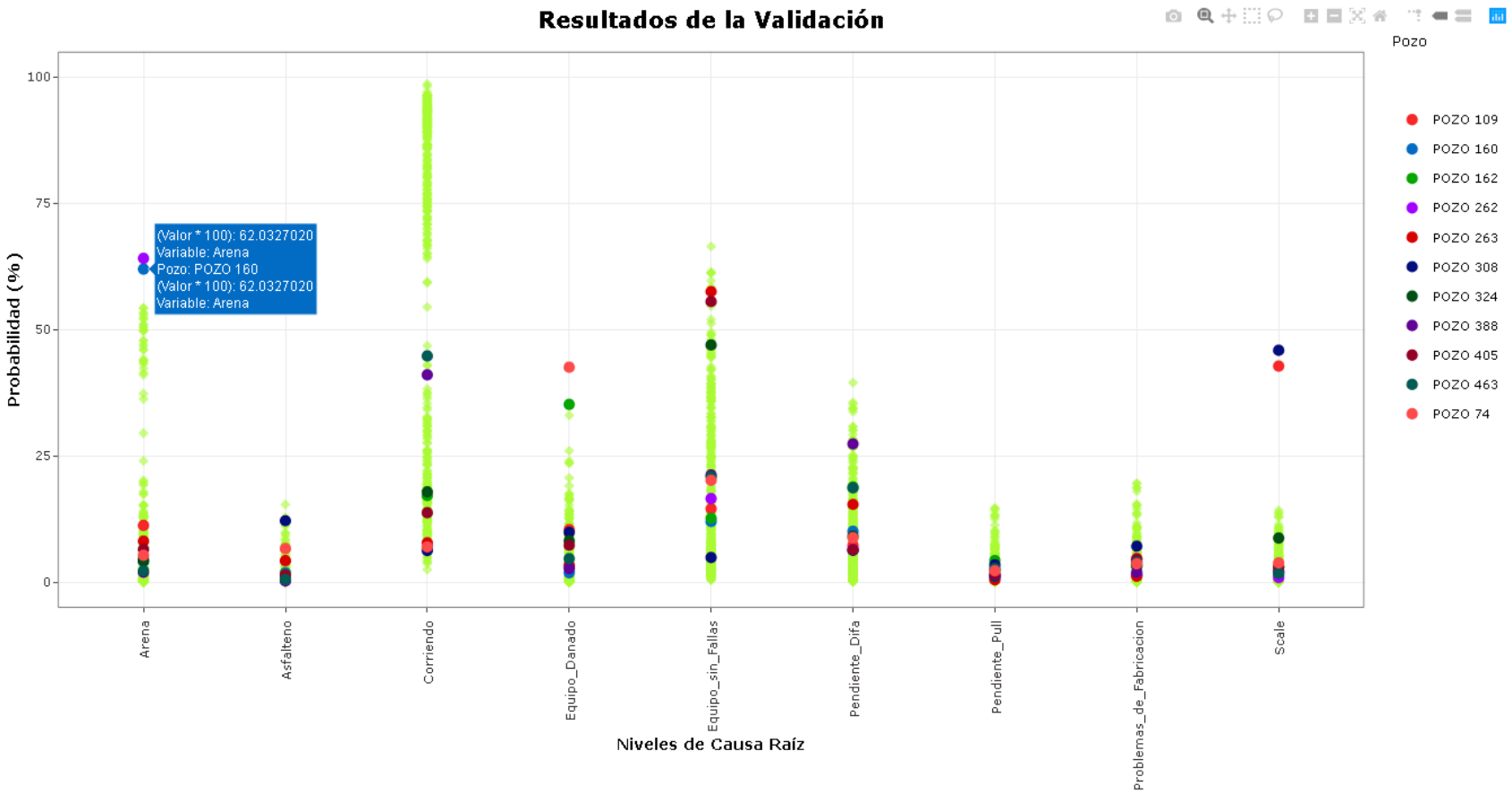
Motor	Tipo Motor	HP Motor	Asentamiento Bomba TVD	Velocidad Fluido	Frecuencia	Eficiencia del Sistema	PIP	BFPD	BSW	TDH
MOTOR 43	TIPO 2	245	4027.23	4.4	88.77	0.57	379.29	8577	0.82	216.82
MOTOR 9	TIPO 2	115	7695.15	5.78	58.49	0.46	1704.17	955	0.73	4650.69
MOTOR 15	TIPO 2	400	2952.13	5.73	82.19	0.35	1905.69	5670	0.87	711.77
MOTOR 20	TIPO 7	115	2325.12	3.03	77.98	0.63	2158.45	4627	0.86	218.73
MOTOR 43	TIPO 2	245	5427.48	2.07	34.85	0.37	754.47	1925	0.94	2809.46
MOTOR 8	TIPO 2	100	5713.84	5.08	72.09	0.78	938.58	760	0.8	2955.77
MOTOR 20	TIPO 6	107	7146.6	3.14	84	0.79	1122.22	1822	0.94	5476.55
MOTOR 12	TIPO 2	160	5612.16	0.09	43.08	0.41	1128.28	718	0.8	4173.18
MOTOR 28	TIPO 3	225	11983.12	5.41	66.73	0.61	874.71	1752	0.9	9683.97
MOTOR 2	TIPO 7	120	3078.28	2.47	48.03	0.24	906.82	1625	0.94	554.36
MOTOR 12	TIPO 3	160	4590.99	4.39	78.66	0.36	1534.94	727	0.8	3251.35

Ty	IP	API	Tipo Crudo	Profundidades Profundo	Profundidades Somero	Arenas Si	Asfaltenos Si	Erosion Si	Scale Si	Fallas Si (Todos igual a 1)
165	8.92	21	Pesado	0	0	0	0	0	0	1
193	6.34	22	Mediano	0	0	1	0	0	0	1
167	3.07	16	Pesado	0	1	0	0	0	0	1
169	7.03	17	Pesado	0	1	0	0	0	0	1
200	6	16	Pesado	0	0	0	0	0	0	1
164	5.91	30	Liviano	0	0	1	0	0	0	1
197	2.06	13	Pesado	0	0	0	1	0	0	1
167	4.74	24	Mediano	0	0	0	1	0	0	1
235	6.32	13	Pesado	1	0	0	0	0	1	1
151	8.48	21	Pesado	0	1	0	0	0	1	1
199	9.74	27	Mediano	0	0	0	0	1	1	1

Nota. Tabla para ingresar los datos del pozo nuevo en el cual un cliente necesita instalar un equipo ESP de la Compañía A. En este caso se observan los 11 pozos escogidos al azar para poder realizar la validación del algoritmo de este proyecto.

Figura 36.

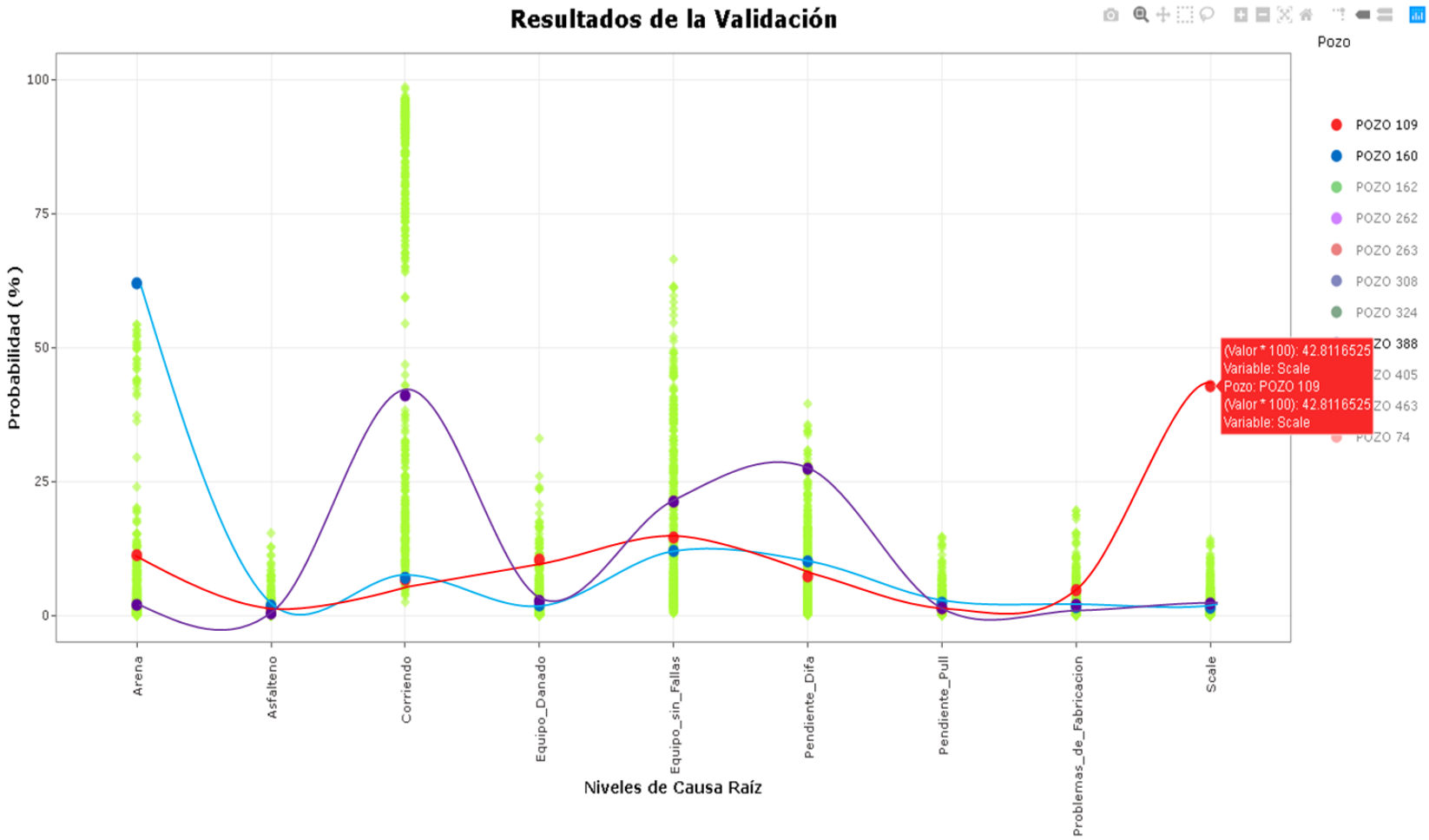
Resultados Generales de la Validación



Nota. En el eje x del gráfico se encuentran cada uno de los niveles de la variable Causa Raíz con sus respectivos valores de probabilidad de ocurrencia (eje y) en cada uno de los 11 pozos (leyenda: cada color hace referencia a un pozo). Los rombos de color verde claro son los resultados obtenidos en el entrenamiento del modelo con el 80%, graficados como plantilla para comparar con los resultados de la validación. Los puntos de colores hacen referencia a las predicciones logradas por el modelo luego de la ejecución del algoritmo con el "Dataframe Validación". En la **Figura 37.**, se exponen tres casos para la validación extraídos desde esta gráfica.

Figura 37.

Análisis de Tres Predicciones



Nota. 1) Pozo 109: demarcado con la línea roja. 2) Pozo 160: demarcado con la línea azul. 3) Pozo 388: demarcado con la línea morada.

Figura 38.

Porcentaje de error entre lo real y la predicción

% Error		
Ecuación		
$\% \, Error = \left\| \frac{Valor\ real\ esperado\ - Valor\ predicción}{Valor\ real\ esperado} \right\| * 100$		
Pozo 109 (Resultado predicción = Scale)		
Valor real esperado (aprox. Al valor máx. de scale)	Valor predicción	%Error
14.21	42.81	201.3
Pozo 160 (Resultado predicción = Arena)		
Valor esperado (aprox. Al valor máx. de arena)	Valor predicción	%Error
54.29	62.03	14.3
Pozo 388 (Resultado predicción = Corriendo)		
Valor esperado (dentro del rango de valores altos > 68%)	Valor predicción	%Error
68	41.06	39.6
Valor predicción más alto en el pozo 388	Segundo valor predicción más alto en el pozo 388	%Error
41.06	27.38	33.3

Nota. El error que comete el modelo predictivo al evaluar la información que se utiliza en la validación, es el desfase que se presenta en un mismo nivel de la variable "Causa Raíz", entre el valor de probabilidad más alto que se alcanza con los resultados de la validación (pozos de la validación) y el valor de probabilidad más alto que se alcanza con los resultados del entrenamiento del modelo (419 pozos del 80%). Esto se puede observar gráficamente en la **Figura 37.,** donde tomando el pozo 109 como ejemplo y el nivel "Scale", se observa lo alejado que está el resultado de la predicción (punto rojo con valor de probabilidad de 42.81%), de los resultados del 80% (rombo verde claro con valor más alto de probabilidad 14.2%). Siguiendo este proceso se calculó el porcentaje de error para cada uno de los tres pozos estudiados en la validación.

3.4.1. Pozo 109

La línea de color rojo representa la distribución de la probabilidad que predijo el modelo para este pozo, donde se resalta que el punto más alto (con una probabilidad de 42.8%) indicó que este pozo tiene una alta probabilidad de fallar por "Scale", con las características y componentes seleccionados para instalar que se observan en la **Tabla 7.** Comparando este resultado con los rombos color verde claro del nivel "Scale", es un resultado que no se presentó en ninguno de los 419 pozos del 80% y que es mucho mayor al valor más alto, por lo cual se estima un error del 201.3%.

Tabla 7.

Información del Pozo 109

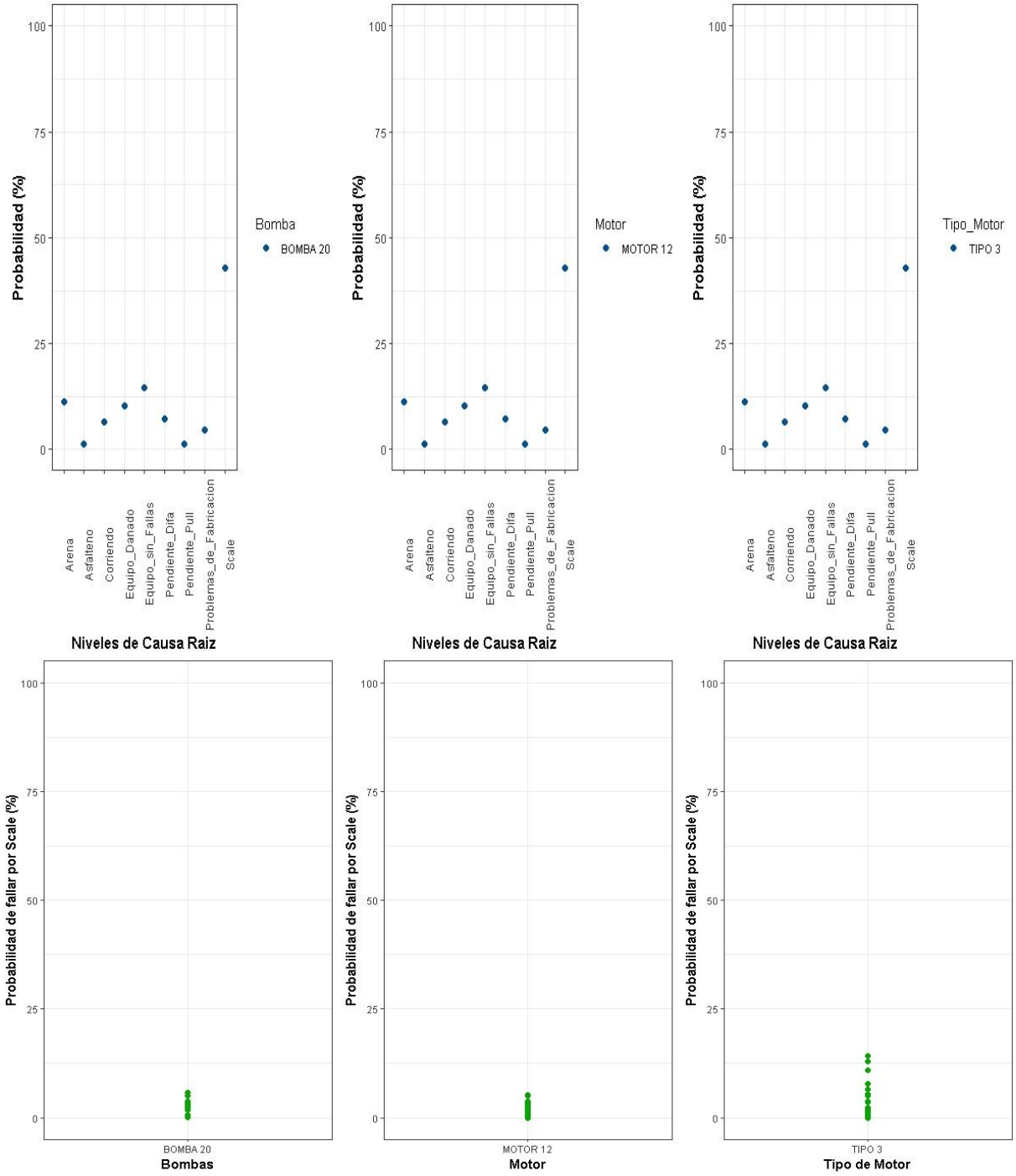
Grupo	Variable	Valor	Unidades
Componentes	Bomba	BOMBA 20	-
	Motor	MOTOR 12	-
	Tipo de Motor	TIPO 3	-
Datos de Diseño	Etapas	100	-
	HP Motor	160	HP
	Asentamiento Bomba TVD	4590.99	ft
	Velocidad Fluido	4.39	ft/s
	Frecuencia	78.66	Hz
	Eficiencia del Sistema	0.36	-
Datos del Match	PIP	1534.94	psi
	BFPD	727	BOPD
	BSW	0.8	-
	TDH	3251.35	ft
Condiciones Especiales de Campo	Ty	199	°F
	IP	9.74	-
	API	27	°
	Tipo Crudo	Mediano	-
	Arenas	No	-
	Asfaltenos	No	-
	Erosión	Si	-
	Scale	Si	-

Nota. Características del pozo y componentes del equipo ESP seleccionados para instalar.

3.4.1.i. Comparación de la predicción por componentes. En la **Figura 39.**, se muestran dos grupos de gráficos, cada uno de a tres gráficos que corresponden a la bomba, el motor y el tipo de motor que se desean instalar en este pozo. En la parte superior, los tres gráficos corresponden a los resultados de la validación graficados para cada uno de los componentes, y en la parte inferior se encuentran graficados los resultados del entrenamiento, con cada uno de los componentes en función del nivel "Scale" (nivel con mayor probabilidad). Se compararon los resultados de la predicción (gráficos de la parte superior) con los resultados del entrenamiento (gráficos de la parte inferior), componente a componente, con el fin de poder observar cómo se han comportado históricamente estos componentes (que se desean instalar) en función de la falla por scale, y así poder obtener una respuesta más sólida. La distribución de probabilidad que se observa en los gráficos de la parte superior demuestra la alta probabilidad que tiene cada componente de fallar y al compararla con los resultados del entrenamiento (gráficos de la parte inferior), se obtuvo que la Bomba 20, el Motor 12 y el tipo de motor Tipo 3, tiene una alta probabilidad de fallar por "Scale" si se instalan en el pozo 160, pero también se observó que históricamente estos tres componentes no presentan probabilidades de fallar por scale mayores al 20%, por lo cual la respuesta debe basarse en ambos gráficos si se desea mayor exactitud.

Figura 39.

Comparación Gráfica de Resultados (Pozo 109)



Nota. Comparación de gráficos que permite observar el comportamiento que han tenido históricamente los componentes del equipo a instalar. Permite llegar a una respuesta sobre la aplicabilidad de los equipos más robusta.

3.4.1.ii. Respuesta y recomendación. Teniendo en cuenta que es un valor cercano al 50% de probabilidad de que el equipo que se instale presente fallas por “Scale”, y luego de que se realizara la comparación, la respuesta es: Se recomienda hacer la instalación de la Bomba 20, el Motor 12 y el tipo de motor Tipo 3, ya que a pesar de tener una alta probabilidad de fallar por scale, la historia muestra que estos componentes han presentado buenos resultados en pozos con presencia de scale, estimándose un error de 201.3%. Por esta razón se acepta el resultado de la predicción de fallar por “Scale” pero sin aceptar el valor tan alto que tomó la predicción, por lo cual se debe realizar una adecuación en el material de los componentes para que estos soporten las condiciones, y a su vez realizar una recomendación al cliente para que realice el adecuado tratamiento del scale para prevenir taponamientos que afecten la producción de crudo.

Adicionalmente se recomiendan como otras opciones los componentes mostrados en la **Tabla 8.**

Tabla 8.

Equipos que Mejor se Comportan Bajo Condiciones de Scale

Orden	Bomba	Motor	Tipo de Motor
1	BOMBA 10	MOTOR 13	TIPO 6
2	BOMBA 20	MOTOR 2	TIPO 7
3	BOMBA 26	MOTOR 40	TIPO 2
4	BOMBA 29	MOTOR 8	TIPO 4
5	BOMBA 18	MOTOR 9	TIPO 5

Nota. A partir la programación de un gráfico (Probabilidad de fallar por scale vs Componente) donde se iba variando el componente, se logró extraer la información para poder dar una tabla de recomendaciones para el Pozo 109, donde se almacenan en orden descendente, cada uno de los componentes que mejor se comportaron en condiciones de arena, según la historia plasmada en las predicciones de los 419 pozo del 80%.

3.4.2. Pozo 160

La línea de color azul representa la distribución de la probabilidad que predijo el modelo para este pozo, donde se resalta que el punto más alto (con una probabilidad de 62.03%) indicó que este pozo tiene una alta probabilidad de fallar por "Arena", con las características y equipos seleccionados para instalar que se observan en la **Tabla 9.** Comparando este resultado con los rombos color verde claro del nivel "Arena", también se observó un resultado mayor a los resultados de probabilidad de los 419 pozos del 80%, por lo cual se estima un error del 14.3%.

Tabla 9.

Información del Pozo 160

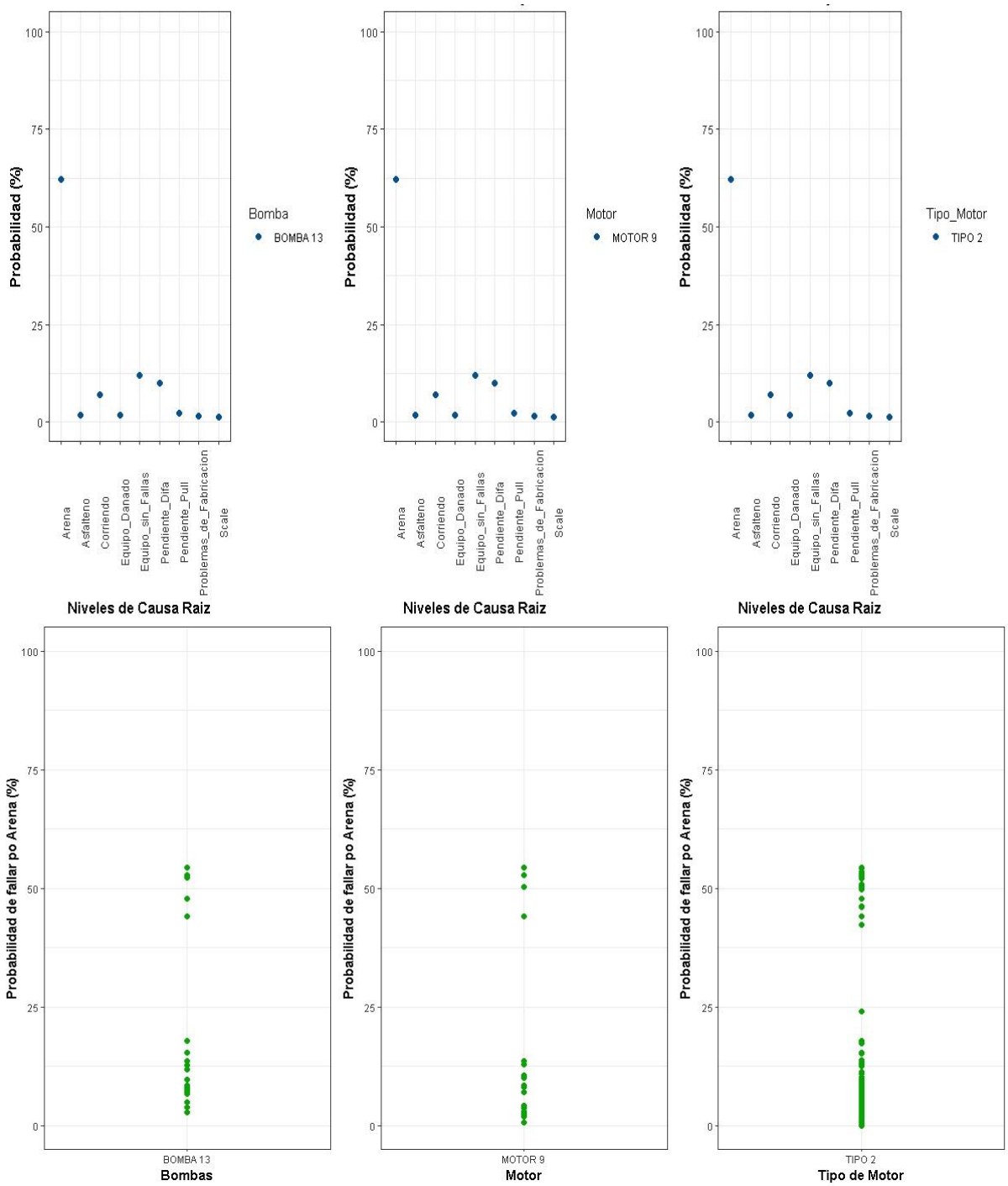
Grupo	Variable	Valor	Unidades
Componentes	Bomba	BOMBA 13	-
	Motor	MOTOR 9	-
	Tipo de Motor	TIPO 2	-
Datos de Diseño	Etapas	150	-
	HP Motor	115	HP
	Asentamiento Bomba TVD	7695.15	ft
	Velocidad Fluido	5.78	ft/s
	Frecuencia	58.49	Hz
	Eficiencia del Sistema	0.46	-
Datos del Match	PIP	1704.17	psi
	BFPD	955	BOPD
	BSW	0.73	-
	TDH	4650.69	ft
Condiciones Especiales de Campo	Ty	193	°F
	IP	6.34	-
	API	22	°
	Tipo Crudo	Mediano	-
	Arenas	Si	-
	Asfaltenos	No	-
	Erosión	No	-
	Scale	No	-

Nota. Características del pozo y componentes del equipo ESP seleccionados para instalar.

3.4.2.i. Comparación de la predicción por componentes. En la **Figura 40.**, se muestran dos grupos de gráficos, los cuales se programaron y se analizaron de la misa forma que se hizo con el pozo 109. En la parte superior, los tres gráficos corresponden a los resultados de la validación graficados para cada uno de los componentes, y en la parte inferior se encuentran graficados los resultados del entrenamiento, con cada uno de los componentes en función del nivel "Arena" (nivel con mayor probabilidad). La distribución de probabilidad que se observa en los gráficos de la parte superior demuestra la alta probabilidad que tiene cada componente de fallar por "Arena", y al compararla con los resultados del entrenamientos (gráficos de la parte inferior), se obtuvo que la Bomba 13, el Motor 9 y el tipo de motor Tipo 2, no solo tiene una alta probabilidad de fallar si se instalan en el pozo 160, sino que también históricamente la Bomba 13 ha presentado fallas por arena en cinco ocasiones, el Motor 9 ha presentado fallas por arena en cuatro ocasiones, y el tipo de motor Tipo 2 en más de 10 ocasiones.

Figura 40.

Comparación Gráfica de Resultados (Pozo 160)



Nota. Comparación de gráficos que permite observar el comportamiento que han tenido históricamente los componentes del equipo a instalar. Permite llegar a una respuesta sobre la aplicabilidad de los equipos más robusta para el Pozo 160.

3.4.2.ii. Respuesta y recomendación. Teniendo en cuenta que es una probabilidad de 62% de que el equipo que se instale presente fallas por “Scale”, y luego de que se realizara la comparación, la respuesta es: No se recomienda hacer la instalación de la Bomba 13, el Motor 9 y el tipo de motor Tipo 2, debido a que no solo es resultado de la predicción indicó la alta probabilidad de fallar por “arena”, sino también la historia demuestra que estos componentes han presentado fallas por arena en repetidas ocasiones. En este caso se recomienda instalar otros modelos de cada componente que tengan históricamente demuestren un mejor comportamiento bajo condiciones similares a las del pozo 160. En la **Tabla 10.**, se observan los componentes en orden de mejor comportamiento en función del nivel “Arena” asociado a las condiciones especiales de campo, de la cual se obtiene las recomendaciones de los nuevos componentes a instalar. Igualmente se debe hacer la recomendación al cliente de realizar una buena estimación de la cantidad de arena y que implementen una acción remedial que evite que, al momento de instalar el equipo, este vaya a presentar daños.

Tabla 10.

Equipos que Mejor se Comportan Bajo Condiciones de Arena

Orden	Bomba	Motor	Tipo de Motor
1	BOMBA 18	MOTOR 2	TIPO 7
2	BOMBA 28	MOTOR 20	TIPO 6
3	BOMBA 10	MOTOR 33	TIPO 2
4	BOMBA 17	MOTOR 17	TIPO 5
5	BOMBA 14	MOTOR 18	TIPO 4

Nota. A partir la programación de un gráfico (Probabilidad de fallar por arena vs Componente) donde se iba variando el componente, se logró extraer la información para poder dar una tabla de recomendaciones para el Pozo 160, donde se almacenan en orden descendente, cada uno de los componentes que mejor se comportaron en condiciones de arena, según la historia plasmada en las predicciones de los 419 pozo del 80%.

3.4.3. Pozo 388

La línea de color morada representa la distribución de la probabilidad que predijo el modelo para este pozo, donde se resalta que el punto más alto (con una probabilidad de 41.06% "Corriendo") indicó que este pozo tiene una alta probabilidad de operar óptimamente sin presentar fallas, con las características y equipos seleccionados para instalar que se observan en la **Tabla 11.** Comparando este resultado con los rombos color verde claro del nivel "Corriendo", se observó un valor promedio de probabilidad en relación con los resultados de probabilidad de los 419 pozos del 80%, por lo cual se determinó que al no estar el resultado entre los puntos de mayor probabilidad (entre 68% y 98%), existe un 33.3% de probabilidad que se presente una falla con el paso del tiempo.

Tabla 11.

Información del Pozo 388

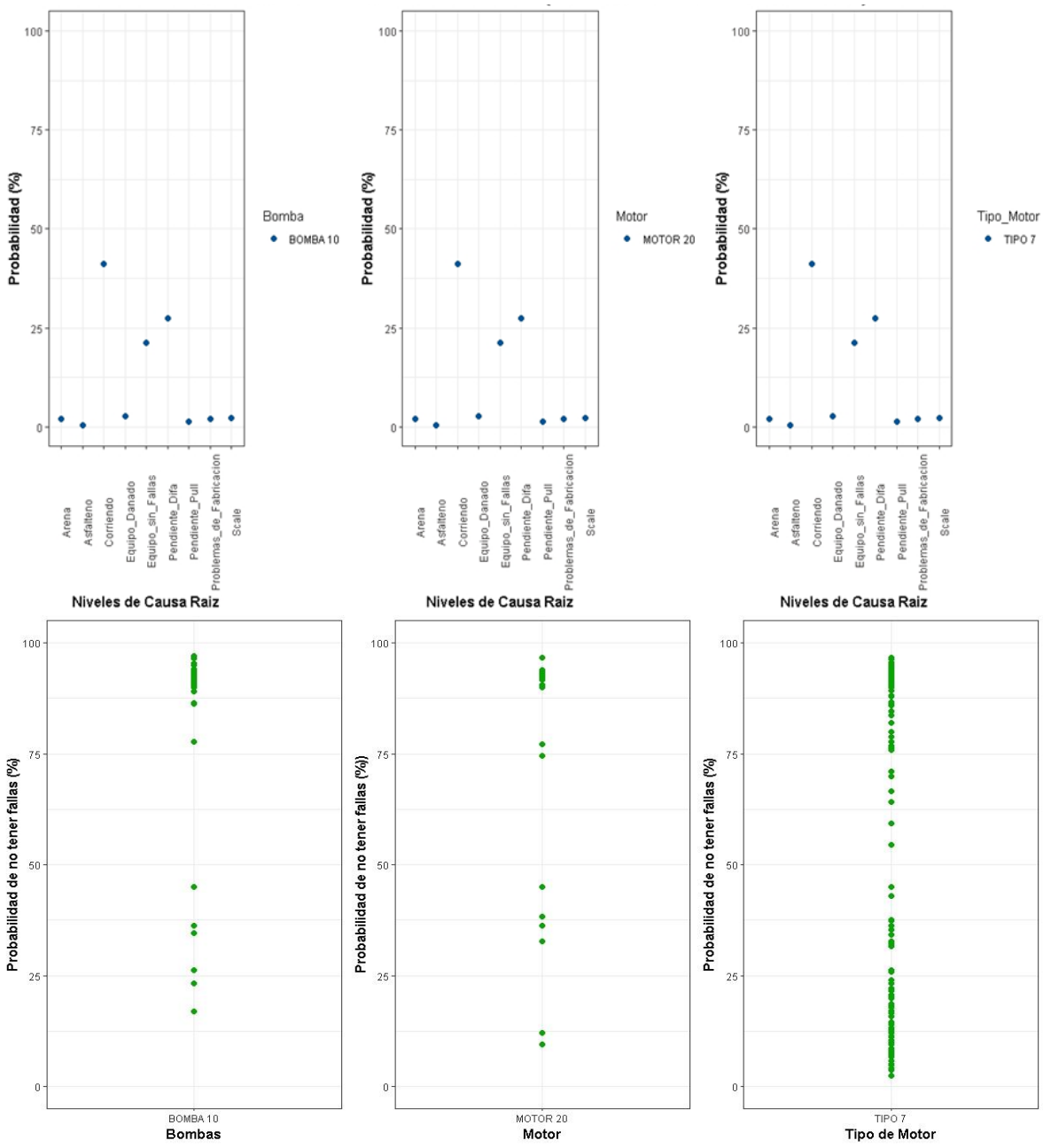
Grupo	Variable	Valor	Unidades
Componentes	Bomba	BOMBA10	-
	Motor	MOTOR 20	-
	Tipo de Motor	TIPO 7	-
Datos de Diseño	Etapas	100	-
	HP Motor	115	HP
	Asentamiento Bomba TVD	2325.12	ft
	Velocidad Fluido	3.03	ft/s
	Frecuencia	77.98	Hz
	Eficiencia del Sistema	0.63	-
Datos del Match	PIP	2158.45	psi
	BFPD	4627	BOPD
	BSW	0.86	-
	TDH	218.73	ft
Condiciones Especiales de Campo	Ty	169	°F
	IP	7.03	-
	API	17	°
	Tipo Crudo	Pesado	-
	Arenas	No	-
	Asfaltenos	No	-
	Erosión	No	-
	Scale	No	-

Nota. Características del pozo y componentes del equipo ESP seleccionados para instalar.

3.4.3.i. Comparación de la predicción por componentes En la **Figura 41.**, se muestran dos grupos de gráficos, los cuales se programaron y se analizaron de la misa forma que se hizo con los pozos anteriores. En la parte superior, los tres gráficos corresponden a los resultados de la validación graficados para cada uno de los componentes, y en la parte inferior se encuentran graficados los resultados del entrenamiento, con cada uno de los componentes en función del nivel "Corriendo" (nivel con mayor probabilidad). La distribución de probabilidad que se observa en los gráficos de la parte superior demuestra la alta probabilidad que tiene cada componente de no presentar fallas y al compararla con los resultados del entrenamiento (gráficos de la parte inferior), se obtuvo que la Bomba 10, el Motor 20 y el tipo de motor Tipo 7, tienen una alta probabilidad de no presentar fallas si se instalan en el pozo 160. Los gráficos que muestran el comportamiento histórico permitieron observar que la mayoría de los puntos graficados en cada componente se encuentran por encima del 75% de probabilidad, lo que garantiza un buen comportamiento de estos si se realiza la instalación del equipo ESP en el pozo 388.

Figura 41.

Comparación Gráfica de Resultados (Pozo 388)



Nota. Comparación de gráficos que permite observar el comportamiento que han tenido históricamente los componentes del equipo a instalar. Permite llegar a una respuesta sobre la aplicabilidad de los equipos más robusta para el Pozo 160.

3.4.3.ii. Respuesta y recomendación. Teniendo en cuenta que es una probabilidad de 42% de que el equipo que se instale no presente fallas, y luego de que se realizara la comparación, la respuesta es: Se recomienda hacer la instalación de la Bomba 10, el Motor 20 y el tipo de motor Tipo 7, debido a que todo resultado de la predicción que indique que el valor más alto es del nivel "Corriendo", con un valor superior al 40% de probabilidad, se toma como un pozo ideal para la instalación del equipo y se espera que los resultados durante la operación sean óptimos. Esta respuesta se basó en el comportamiento histórico de 419 pozos del 80%, donde se evidenció que la respuesta "Corriendo" es el nivel de la variable "Causa_Raíz" que más se repite, ya que el 57% de los equipos instalados en cada pozo se encuentran corriendo óptimamente.

4. CONCLUSIONES

La construcción de un *dataframe* eficiente se consolida como el resultado más importante para poder iniciar la implementación de metodologías de *Machine Learning* en RStudio. Este debe contener información verídica y estructurada para garantizar un menor esfuerzo a la hora de programar el algoritmo y los más importante, garantizar buenos resultados predictivos.

El algoritmo que se desarrolla en este proyecto tiene la capacidad de evaluar estadísticamente el 100% de la información del *dataframe*, identificar los 25 predictores óptimos, dividir el *dataframe* en 80% entrenamiento y 20% test, ejecutar el modelo predictivo (para el entrenamiento, el test y la validación), y visualizar gráficamente los resultados de las predicciones, lo cuales permiten extraer toda la información para llegar a la respuesta sobre la aplicabilidad de los equipos ESP.

La selección de los 25 predictores para el modelo *Random Forest* incluye las variables arenas, asfaltenos, erosión y scale, que son parte de las variables del grupo "Condiciones Especiales de Campo", lo que se traduce como un buen resultado debido a que el estudio está centrado en las afectaciones que generan dichas condiciones a los equipos ESP.

La comparación entre los resultados de la predicción de los 103 pozos de conjunto test y los valores originales de estos 103 pozos en el "Dataframe Principal", permite comprobar la funcionalidad del algoritmo, el cual tuvo la capacidad de predecir valores distintos a los originales.

Para llegar a la respuesta sobre la aplicabilidad de los equipos ESP en pozos nuevos, se debe seguir el mismo análisis conjunto de tres factores como se realiza en los Pozos 109, 160 y 388. El primero es el resultado gráfico que representa la distribución de probabilidad de cada pozo. El segundo es la comparación entre el resultado gráfico de la predicción y los gráficos que representan los comportamientos históricos de los componentes Bomba, Motor y Tipo de Motor. Y el tercero es la recomendación de los componentes que mejor se comportan bajo condiciones de Arena, Asfalteno, Erosión y Scale.

El algoritmo predice probabilidades a partir de los valores de entrada, y tiene gran influencia especialmente por los valores que toman los niveles de variables cualitativas binarias ("si" y "no") como Arena, Asfalteno, Scale y Erosión, lo que ocasiona que los resultados de las validaciones presenten porcentajes de error mayores al 50%.

En los resultados de la validación, donde la predicción presenta porcentajes de error superiores al 50%, se descarta el porcentaje alto de probabilidad que se obtiene de la predicción, pero no se descarta el nivel de la variable "Causa Raíz" que se obtiene como resultado de la predicción, como sucede en el Pozo 109, donde la recomendación indica rechazar el valor de 42.8% e instalar los componentes inicialmente seleccionados (Bomba 20, Motor 12, y tipo de motor Tipo 3), pero sugiriendo realizar adecuaciones al material de los componentes por parte de la Compañía y que el cliente realice un tratamiento de scale en el pozo para prevenir taponamientos o fracturas en el equipo ESP.

El porcentaje de error en casos donde el resultado de la predicción indica una alta probabilidad de fallar a causa de "Scale", tiende ser elevado con respecto a los demás niveles de la variable "Causa Raíz", como se observa en el análisis del Pozo 109, el cual presenta un desfase del 201.3% entre el porcentaje predicho (42.8%) y el máximo porcentaje esperado (14.2%).

El porcentaje de error en casos donde el resultado de la predicción indica una alta probabilidad de fallar a causa de "Arena", es aceptable ya que no presenta desfases mayores al 20%, como se observa en el análisis del Pozo 160, el cual presenta un desfase del 14.3% entre el porcentaje predicho (62.03%) y el máximo porcentaje esperado (54.29), por lo cual se acepta el resultado.

El resultado de la predicción del Pozo 160 indica que la respuesta es "arena" con un porcentaje mayor al 45%. La comparación entre este resultado y el comportamiento histórico de los componentes, en este caso de la Bomba 13, el Motor 9 y el tipo de motor Tipo 2, indica que en varias instalaciones estos componentes presentaron fallas por arena. En estos casos se debe realizar recomendaciones de componentes que históricamente presentaron los mejores resultados operando bajo condiciones de arena. Para este pozo se recomendó no instalar los equipos seleccionados inicialmente, sino realizar la instalación con los componentes Bomba 18, Motor 2 y tipo de motor Tipo 7, lo que probablemente garantiza mejores resultados.

El resultado de la predicción del Pozo 388 muestra cómo es la distribución de probabilidad cuando el resultado de la predicción es "Corriendo". Con un valor de probabilidad mayor al 40% se garantiza que el equipo a instalar va a operar óptimamente, pero se garantizan mejores resultados en casos donde el resultado de la predicción esté entre 68% y 98%, que es el rango de valores más alto de probabilidad en este nivel. Este pozo obtuvo una probabilidad del 42% aproximadamente,

por lo cual no se descarta que alguno de sus componentes pueda llegar a fallar con el paso del tiempo, ya que existe un desfase del 33.3% en relación con el segundo valor más alto de la distribución de probabilidad en este pozo.

BIBLIOGRAFÍA

[1] L.A. Cortés, M.F. Delgado, *Evaluación técnico financiera para el cambio del sistema de levantamiento artifical actual por bombeo por cavidades progresivas con motor de fondo de imanes permanentes en tres pozos de un campo petrolero*, Tesis Pregrado, Facultad de Ingenierías, Fundación Universidad de América, Bogotá, Colombia, 2018, [En línea]. Disponible: https://hdl.handle.net/20.500.11839/6809.

[2] F. Cachumba, *Estudio para la optimización de producción de pozos con bombeo electrosumergible, mediante análisis nodal del campo Cuyabeno*, Tesis Pregrado, Facultad de Ingeniería en Geología y Petróleos, Escuela Politécnica Nacional, Quito, Ecuador, 2017, [En linea]. Disponible: http://bibdigital.epn.edu.ec/handle/15000/18852.

[3] M. Colorado, *Evaluación técnico financiera del rendimiento de los motores de imanes permanente con bombas electrosumergibles de alta eficiencia del Campo A ubicado en la Cuenca Llanos Orientales*, Tesis Pregrado, Facultad de Ingenierías, Fundación Universidad de América, Bogotá, Colombia, 2016, [En línea]. Disponible: https://hdl.handle.net/20.500.11839/514.

[4] Recommended Practice for Sizing and Selection of Electric Submersible Pump Installations, API 11S4, 3ª ed., American Petroleum Institute, Washington D. C., EE. UU., 2002.

[5] F. H. Escobar. (2012). *Fundamentos de Ingeniería de Yacimientos.* (1ª ed.). [En línea]. Disponible en: http://oilproduction.net/files/Libro%20Fundamentos%20de%20Ing%20de%20Yacimientos%20-%20Fredy%20Escobar.pdf.

[6] Schlumberger. (s.f). "Oilfield Glossary en Español. Índice de Productividad". [En línea]. https://www.glossary.oilfield.slb.com/es/Terms/p/productivity_index_pi.aspx#:~:text=Una%20forma%20matem%C3%A1tica%20de%20expresi%C3%B3n,bbl%2Fd%2Fpsi). [Acceso: noviembre 9, 2020].

[7] C. Bánzer. (1996). *Correlaciones Numéricas PVT*, (1ª ed.). [En línea]. Disponible en: http://oilproduction.net/files/Correlaciones%20PVT-Carlos%20Banzer.pdf.

[8] Petroquimex. (25, may, 2017). "Dispersantes de Parafinas y Asfaltenos, una Solución para el Sostenimiento y Optimización en la Producción de Crudo". [En línea]. https://petro_quimex.com/dispersantes-de-parafinas-y-asfaltenos-una-solucion-para-el-sostenimiento-y-optimizacion-en-la-produccion-de-crudo/. [Acceso: noviembre 12, 2020].

[9] M. Crabtree, D. Eslinger, A. Johnson, G. King, P. F. Matt, "La Lucha Contra las Incrustaciones. Remoción y Prevención", 1999, [En línea]. Disponible en: https://www.slb.com/-/media/files/oilfield-review/p30-49.

[10] Adp. (4, mar, 2019). "¿Qué es Machine Learning y cómo funciona?". [En línea]. https://www.apd.es/que-es-machine-earning/#:~:text=Machine%20Learning%20o%20Aprendizaje%20autom%C3%A1tico,de%20datos%20en%20su%20sistema. [Acceso: noviembre 10, 2020].

[11] H. Ahumada, *et al.*, *Extensión de Métodos Modernos de Aprendizaje Automátizado y Aplicaciones*, Objeto de Conferencia, Ciencias Informáticas, Universidad Nacional de la Plata, La Plata, Argentina, 2011, [En línea]. Disponible: http://sedici.unlp.edu.ar/handle /10915/19971.

[12] BBVA. (8, nov, 2019). "Machine Learning: ¿Qué es y como funciona?". [En línea]. https://www.bbva.com/es/machine-learning-que-es-y-como-funciona/. [Acceso: noviembre 10, 2020].

[13] H. Asurza. (2006). *Glosario Básico de Términos Estadísticos.* [En línea]. Disponible en: https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib0900/Libro.pdf.

[14] Máxima Formación. (s.f.). "Data Science. Qué es R Software". [En línea]. https://www.maximaformacion.es/blog-dat/que-es-r-software/. [Acceso: noviembre 11, 2020].

[15] RStudio. (s.f.). "RStudio. Toma el control de tu código en R". [En línea]. https://rstudio.com/products/rstudio/. [Acceso: noveimbre 12, 2020].

[16] A. Santana. (2012). *Introducción al Entorno Estadístico.* [En línea]. Disponible en: https://docplayer.es/70903696-Introduccion-al-entorno-estadistico-angelo-santana.html.

[17] CienciadeDatos.net. (s.f.). "Ciencia de Datos, Estadística, Machine Learning y Programación". [En línea]. https://www.cienciadedatos.net/. [Acceso: agosto 30, 2020].

[18] CienciadeDatos.net. (Abril, 2018). "Machine Learning con R y Caret" [En línea]. https://www.cienciadedatos.net/documentos/41_machine_learning_con_r_y_caret. [Acceso: septiembre 10, 2020].

[19] Galería de Gráficos R. (s.f.). "Correlograma" [En línea]. https://www.r-graph-gallery.com/correlogram.html. [Acceso: noviembre 27, 2020].

[20] CienciadeDatos.net. (Febrero, 2017). "Árboles de decisión, random forest, gradient boosting y C5.0" [En línea]. https://www.cienciadedatos.net/documentos/33_arboles_de_prediccion_bagging_random_forest_boosting. [Acceso: diciembre 2, 2020].

ANEXOS

ANEXO 1

HOJA DE DATOS DE DISEÑO DE ESP

APPENDIX B—ESP DESIGN DATA SHEET

Operator: _____ Lease: _____ Well: _____
Location: _____ Field: _____ Resvr: _____
Prepared by: _____ Company: _____
Date: _____

WELLBORE GEOMETRY
Deviation: Yes | No (if yes, include deviation survey and indicate following depths as TD or MD)
Prod. interval Top: _____ ft | m Bottom: _____ ft | m
Casing min. ID: _____ in. | cm Wt: _____ lb/ft | kg/m PBTD: _____ ft | m
Liner min. ID: _____ in. | cm Grade: _____ lb/ft | kg/m TOL: _____ ft | m
Tubing OK: _____ in. | cm Wt: _____ lb/ft | kg/m
Grade: _____ Thread: _____
Tubing ID: _____ in. | cm Burst: _____ psi | bar | kPa
Limiting factors in wellbore (y-tools, packers, etc)?

SURFACE INFORMATION
Flowline ID: _____ in. | cm Length: _____ ft | m Elevation: _____ ft | m
Separator/Wellhead pressure: _____ psig | bar | kPa Temperature: _____ °F | °C
Casing pressure: _____ psig | bar | kPa Vented? _____ Yes | No
Primary power: _____ Volts Frequency _____ Hz
Amperage Limitations? _____

FLUID PROPERTIES
Oil specific gravity: _____ Water specific gravity: _____
Paraffin? _____ Asphaltenes? _____ Scaling? _____
Detailed information on above: _____

Gas specific gravity: _____ H_2S content: _____ ppm CO_2 content: _____ ppm
Water cut: _____ % CLR | GOR: _____ scf/bbl | m^3/m^3
Sand? _____ Yes | No Shape of sand grains: Round | Angular
Bubble pt pressure: _____ psia | bar | kPa

PVT & Viscosity Data

P (psia | bar | kPa)	T (°F | °C)	B_o (bbl/stb | m^3/m^3)	B_g (bbl/stb | m^3/m^3)	R_s (cf/scf | m^3/m^3)	μ_{od} (cP | SSU)	μ_o (cP | SSU)

Emulsion Viscosity Correction Factors _____ Inversion point _____ % water
Water cut: _____ _____ _____ _____ _____
Factor: _____ _____ _____ _____ _____

INFLOW CHARACTERISTICS
Test datum MD: _____ ft | m TVD: _____ ft | m
Static pressure: _____ psig | bar | kPa Temperature: _____ °F | °C
Test rate (oil | liq): _____ _____ _____ _____ bpd | m^3/d
Test pressure: _____ _____ _____ _____ psig | bar | kPa
Productivity Index _____ bpd/psi | $m^3/d/kPa$

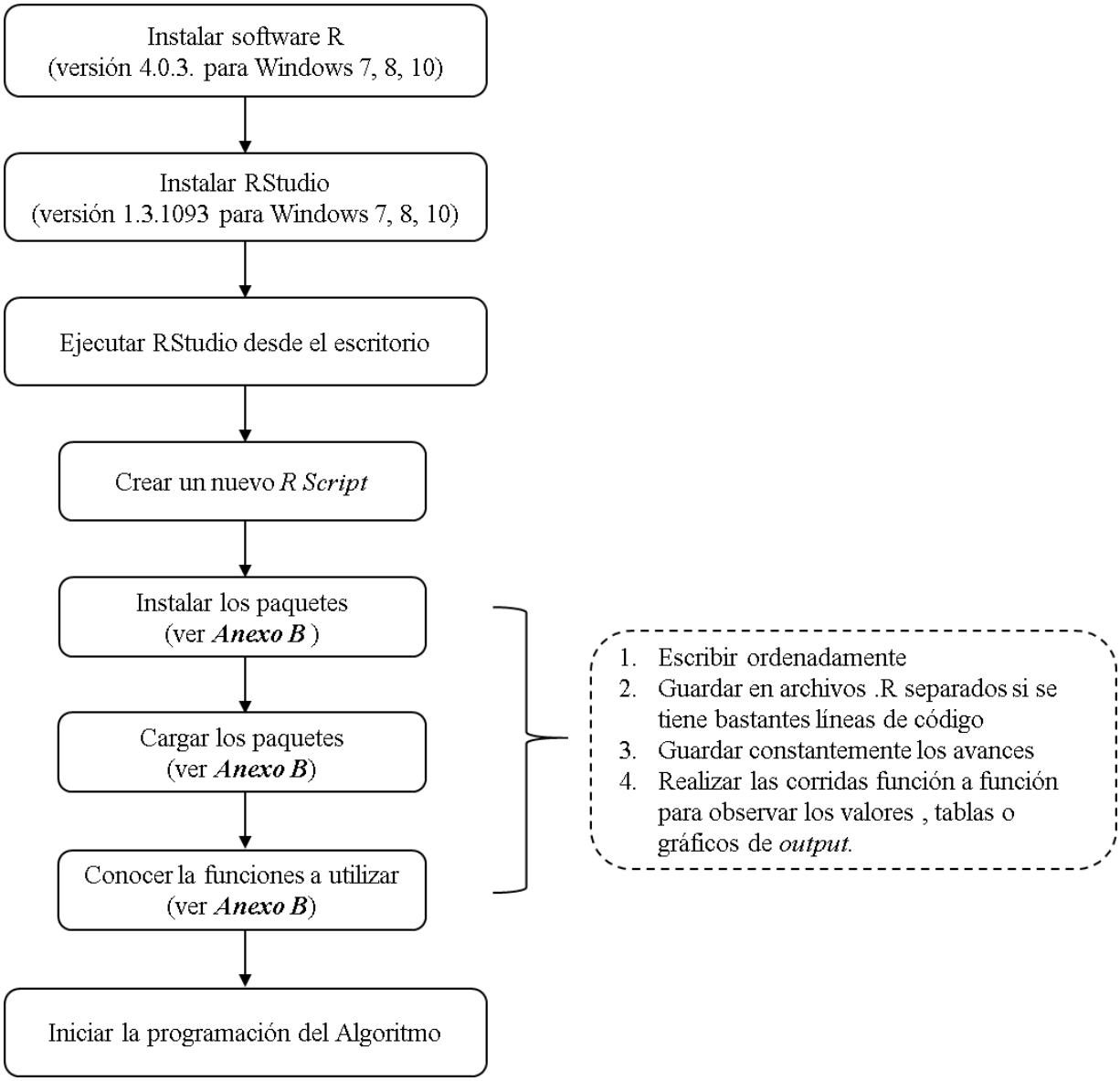
DESIGN CRITERIA
Desired flow rate (oil | liq): _____ bpd | m^3/d
Desired pump intake pressure: _____ psig | bar | kPa
Minimum fluid over pump: _____
Desired pump setting depth MD: _____ ft | m TVD: _____ ft | m
Switch gear/trans. rating: _____ KVA Min Hz: _____ Max Hz: _____
Available voltage taps: _____
Comments: _____

25

Nota. Tomado de: Recommended Practice for Sizing and Selection of Electric Submersible Pump Installations, API 11S4, 3ª ed., American Petroleum Institute, Washington D. C., EE. UU., 2002.

ANEXO 2

DIAGRAMA DE FLUJO PARA EL USO DE RSTUDIO



Nota. Serie de pasos básica para la descarga, instalación y uso del software RStudio. Las ejecuciones se realizan seleccionando las líneas que se desee con clic izquierdo sostenido y usando el comando Ctrl+Enter. Cada vez que se haya guardado y finalizado un trabajo, cuando se desee abrir nuevamente un archivo, se debe ejecutar todas las líneas de código nuevamente, ya que el software no almacena la información una vez finalizada la sesión. Elaborado por J.S. Andrade

ANEXO 3

PAQUETES Y FUNCIONES UTILIZADAS EN RSTUDIO

OPERADORES	**DESCRIPCIÓN**
<-	Asigna funciones a un objeto.
%>%	Concatena funciones del paquete dplyr.
+, -, *, /, ^	Operadores matemáticos.
= =, !=, <, >, <=, >=, & (and), | (or)	Operadores Lógicos.
if (*condicion*) {*operaciones* }	Estructura para crear condicionales.
for (generadores de rango) {*operaciones* }	Estructura para crear bucles de repetición.
$	Permite extraer información de un *dataframe* o de un resultado de un modelo (dataframe$variable).
FUNCIONES	**DESCRIPCIÓN**
install.packages (*"paquete"*)	Instala en Rstudio los paquetes que no vienen por defecto y que sean requeridos por el usuario.
library (*"librería"*)	Activa las librerías que contienen los paquetes y que sean requeridos por
setwd (*"ruta"*)	Establece una ruta a un carpeta del computador para importar y
read_excel (*"archivo excel"*)	Carga archivos directamente desde excel. Ideal para cargar
write.table (*"dataframe", sep = ",", file = "nombre.txt"*)	Permite exportar *dataframes* en formato de texto.
view (*"dataframe"*)	Permite observar el *dataframe* en la zzona del *Rscrip* .
select (*"rango"*)	Permite extraer columnas de un dataframe.
if_else (*"condición", "true", "false"*)	Permite programar condiciones sin necesidad de una estructura de
ggplot (*"Datos", aes(x="" , y="")*) + geom_point (*"Argumentos Estéticos"*) + labs (*"Títulos del gráfico"*) + *"otras funciones que mejoran la estética del gráfico"*	Grafica datos de un *dataframe* según desee el usuario: puntos, barras, columnas, histogramas, tortas, *boxplots* , entre otros. Se pueden agregar otras funciones que mejoren la estética del gráfico.
ggarrange (*"# de ggplot"*)	Concatena gráficos de la función *ggplot ( )* para mostrarlos en una sola
ggpairs (*"dataframe"*)	Permite programar un *Correlograma*.
ggplotly (*"gráfcio"*)	Permite transformar un gráfico ggplot ( ) en un gráfico dinámico.
cor.test (*"x = "", y = "", method = "pearson"*)	Realiza un test estadístico y devuelve el *p-value* y el coeficiente de correlación de *Pearson* .
set.seed (*"#"*)	Permite seleccionar datos de forma aleatoria según el número que se le ingrese como argumento
createDataPartition (*y = "dataframe", p = "", ...*)	Permite dividir el dataframe en dos partes, según el porcentaje que se establezca en el argumento *p=""* .
registerDoParallel (*cores = ""*)	Permite paralelizar las corridas de los modelos para que sean más rápidas. Se ingresa el numero de procesadores que tiene el computador

Nota. Elaborado por J.S. Andrade a partir de las funciones utilizadas a lo largo de este proyecto

ANEXO 4

RECOMENDACIONES

Cambiar los valores de las variables Arena, Asfalteno, Erosión y Scale, a valores cuantitativos, para evitar que la predicción se vea influenciada por variables cualitativas binarias.

Utilizar otro modelo predictivo para observar nuevos resultados y compararlo con el modelo *Random Forest* con el fin de determinar realmente cual es el mejor modelo para predecir variables cualitativas.

Aplicar el modelo pasándole la información de nuevos pozos y llevar un control periódicamente del comportamiento de los equipos y las condiciones del pozo, con el fin de poder determinar realmente cual es la efectividad del algoritmo.

Implementar líneas de código tanto en la ejecución del modelo con el conjunto test (20%) como en la ejecución de la validación, que tengan la capacidad de almacenar los mismos resultados que se mostraron gráficamente en este proyecto, con el fin de que el algoritmo final muestre la respuesta en una nueva ventana, donde se evidencie fácilmente la respuesta sobre la aplicabilidad y la recomendación; entre una y cinco recomendaciones de los componentes que mejor se comportarían según los valores de entrada del pozo a predecir.