UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Machine Learning en la predicción de propiedades PVT

TESIS

Que para obtener el título de

Ingeniero Petrolero

P R E S E N T A

Bely Iván Melgar Nieto

DIRECTOR DE TESIS

Dr. Víctor Leonardo Teja Juárez

Ciudad Universitaria, Cd. Mx., 2022

Dedicatoria

A mi madre Diana y mi padre Belisario.

Gracias por su apoyo incondicional, por nunca dejarme solo y ayudar a formarme como la persona que soy el día de hoy.
Los amo.

Agradecimientos

A mis abuelos Belisario, Lidia, Felipe y Rosario por todo su cariño a lo largo de mi vida y estar siempre presentes.

A Dani, Arturo, Fer, Ulises, Roy y toda la crew por su amistad incondicional y apoyo.

A mi director de tesis, el Dr. Víctor Teja por su apoyo, tiempo, atención y observaciones para poder llevar a cabo este proyecto.

A los miembros del jurado: Dra. Paulina Gómora, Dr. Iván Guerrero, Dr. Rodolfo Camacho y M.C. Luis Loera por sus comentarios y sugerencias durante la revisión de mi tesis.

Al proyecto PAPIIT IA106820 por facilitar los recursos computacionales que ayudaron en el desarrollo de esta tesis.

A la Fundación Telmex Telcel y Grupo BAL por el apoyo económico brindado durante mis estudios.

A la Escuela Nacional Preparatoria Plantel 6. "Antonio Caso", la Facultad de Ingeniería y la Universidad Nacional Autónoma de México por todas las amistades, experiencias y conocimientos que me dieron a lo largo de mi vida académica.

Índice general

Índice de figuras

Índice de tablas

Nomenclatura

p	Presión
V	Volumen
T	Temperatura
R_s	Relación de solubilidad
p_b	Presión de burbuja
B_α	Factor de volumen de formación de la fase α
ρ_α	Densidad de la fase α
γ_α	Densidad relativa de la fase α
C_α	Coeficiente de compresibilidad isotérmica de la fase α
μ_α	Viscosidad de la fase α
μ_{oD}	Viscosidad del aceite sin gas disuelto (muerto)
z	Factor de desviación del gas
R	Constante universal de los gases ideales
M	Masa molecular
M_a	Masa molecular aparente
y	Fracción mol de los componentes de la fase gas
RGA	Relación gas-aceite
$^\circ API$	Densidad API
SCF	Pies cúbicos a condiciones estándar (por sus siglas en inglés)
STB	Barriles a condiciones de tanque de almacenamiento (por sus siglas en inglés)
rb	Barriles a condiciones de yacimiento (por sus siglas en inglés)
rcf	Pies cúbicos a condiciones de yacimiento (por sus siglas en inglés)
cP	Centipoise
$^\circ F$	Grados Farenheit
psi	Libras por pulgada cuadrada (por sus siglas en inglés)
$psia$	Libras por pulgada cuadrada absolutas (por sus siglas en inglés)

ML	Aprendizaje de máquina *(Machine learning)*
AI	Inteligencia artificial *(Artificial Intelligence)*
$Extra\ Trees$	Algoritmo de *Extremely Randomized Trees*
SVR	Regresión con vectores de soporte *(ν-Support Vector Regression)*
ANN	Red neuronal artifical *(Artificial Neural Network)*
KNN	Algoritmo de *K-Nearest Neighbors*
R^2	Coeficiente de correlación
MAE	Error medio absoluto
μ	Media aritmética
$d.e.$	Desviación estándar
min	Mínimo
max	Máximo
∂	Derivada parcial
$\sum$	Suma
$p_b \rightarrow B_{ob}$	Predicción de Bob a partir de p_b generada

Sufijos

α	Fase o, g, w
o	Fase líquida (aceite)
w	Fase líquida (agua)
g	Fase gaseosa
gd	Gas disuelto
b	Condiciones de presión de burbuja
c	Condición crítica
r	Condición reducida
pc	Condición pseudocrítica
pr	Condición pseudoreducida
ce	Condiciones estándar de presión y temperatura
p, T	Condiciones de presión p y temperatura T
i	Componente i
n	Número de componentes

Resumen

El *Machine Learning* (ML) o aprendizaje de máquina es una rama de la inteligencia artificial (*artificial intelligence*, AI), que usa datos históricos para estimar el comportamiento de una variable mediante distintas técnicas de manejo de datos y algoritmos, además de mejorar automáticamente su eficiencia a medida que adquiere más experiencia. Este trabajo tiene como objetivo, desarrollar una herramienta computacional que aplique el ML en la estimación de propiedades PVT (tradicionalmente determinadas en laboratorios especializados o mediante correlaciones empíricas), específicamente la presión de burbuja y el factor de volumen de formación del aceite a la presión de burbuja. Para lograr esto, se entrenaron seis algoritmos de ML con un set de datos perteneciente a campos de distintas regiones del mundo compuesto por más de 400 valores para cada una de las cuatro propiedades (temperatura de yacimiento, densidad API, densidad relativa del gas y relación de solubilidad). Los algoritmos comprenden variantes de árboles de decisión, máquina de vectores de soporte, redes neuronales artificiales y *Nearest Neighbors*. Para evaluar el desempeño de los algoritmos, se compararon los resultados generados contra cinco de las correlaciones empíricas más usadas en la ingeniería petrolera (M. B. Standing, 1977, Glasø, 1980, Al-Marhoun, 1988, Dokla y Osman, 1992 y Petrosky y Farshad, 1993). De acuerdo con los resultados obtenidos mediante el conjunto de validación, se logró estimar ambas propiedades PVT (presión de burbuja y factor de volumen de formación del aceite a la presión de burbuja) con un grado de precisión que igualó (y superó) al de las correlaciones. Esto abre la posibilidad de implementación del ML como una herramienta robusta para la estimación de propiedades PVT.

Abstract

Machine Learning (ML) is a branch of artificial intelligence (AI), which uses historical data to predict the behavior of a variable by handling different data techniques and algorithms which automatically improve efficiency as more experience is acquired. The objective of this work is to develop a computational tool which uses ML in the prediction of PVT properties (traditionally determined in specialized laboratories or estimated by empirical correlations), specifically the bubble point pressure and the oil formation volume factor at bubble point. To achieve this, six ML algorithms were trained with a dataset comprised with information from different regions of the world (with more than 400 values for each of the following properties: API gravity, reservoir temperature, gas specific gravity and solution gas-oil ratio). The algorithms include variants of Decision Trees, Support Vector Machines, Artificial Neural Networks and Nearest Neighbors. To evaluate performance, the results were compared against five of the most used correlations in petroleum engineering (M. B. Standing, 1977, Glasø, 1980, Al-Marhoun, 1988, Dokla and Osman, 1992 and Petrosky and Farshad, 1993). The results from the validation dataset show that the two PVT properties (bubble point pressure and oil formation volume factor at bubble point) were predicted with an equal (or greater) degree of precision to the correlations used. This opens the possibility to implement ML as a powerful tool to estimate PVT properties, with the advantage that the more information the model has, the estimates accuracy will increase.

Capítulo 1

Introducción

En este primer capítulo se muestra el estado del arte actual respecto al *Machine Learning* (ML) o aprendizaje de máquina y su relación con la ingeniería petrolera, haciendo énfasis en la estimación de propiedades PVT. Seguido de esto, se explican los objetivos de esta tesis y, se presenta brevemente la estructura de este trabajo.

1.1. Estado del arte

La inteligencia artificial (artificial intelligence, AI) se basa en la hipótesis de que el pensamiento de alguna manera puede implantarse en una máquina, en otras palabras, que puede codificarse. De acuerdo con Mueller y Massaron, 2021, esta idea ha sido perseguida por distintas culturas desde hace siglos. Personajes como Thomas Hobbes, Gottfried Leibniz, o incluso René Descartes han hablado del potencial que implicaría trasladar todos los pensamientos humanos a un lenguaje matemático.

Este problema ha sido explorado extensamente en las últimas décadas, desde lo explicado por Alan Turing, en su trabajo llamado "Computing Machinery and Intelligence" publicado en 1950, se explora la idea de construir una máquina que pueda imitar el comportamiento humano y de como podría determinarse si en efecto, la máquina es capaz de pensar (Mueller y Massaron, 2021). Ese puede considerarse como el nacimiento de la inteligencia artificial tal y como se concibe actualmente.

La inteligencia artificial está compuesta de varias áreas de conocimiento, una de ellas y que es esencial es el ML. Como explican Shobha y Rangaswamy, 2018, el ML surge "como respuesta a la pregunta de cómo construir un programa computacional a partir de datos históricos que pueda resolver un problema dado, además de tener la capacidad de mejorar automáticamente la eficiencia del programa mediante la experiencia".

El ML involucra diversas disciplinas, ciencias de la computación, matemáticas, estadística, probabilidad, ingeniería, ciencias de datos, entre otras. No se puede tener un entendimiento completo del ML sin involucrar conceptos de las áreas antes mencionadas.

Actualmente se genera una cantidad inmensa de datos en diversos campos, por ejemplo, cuando una persona usa un dispositivo para acceder a un sitio web como una red social, quedan registrados datos como el tipo de dispositivo, la cantidad de tiempo dentro del sitio entre muchas otras variables. Tan solo en México más del 70 % de la población mayor a seis años tiene acceso a internet. De acuerdo con datos del INEGI, 2021, esto representa más de 84 millones de usuarios generando datos continuamente. Esta información puede ser aprovechada por medio de algoritmos de ML porque toma como punto de partida los datos y "aprende" de estos, de su estructura y patrones.

Existe una gran cantidad de ejemplos donde el ML ha sido implementado exitosamente, tanto en actividades del día a día como en problemas con una gran complejidad. Cuando el lector, que muy probablemente disponga de un teléfono celular, le habla a éste para preguntarle acerca del pronóstico meteorológico, el dispositivo es capaz de "entender" estas órdenes y ejecutar lo que el usuario está pidiendo, esta tarea que hoy en día parece tan simple es un ejemplo donde se aplica el ML para permitir la comunicación entre un humano y una máquina.

Al momento de abrir una aplicación de *streaming* tal como *Amazon* o *Netflix* se muestran recomendaciones personalizadas, estas recomendaciones tienen detrás un algoritmo de ML que registra los hábitos del usuario y emite sugerencias de acuerdo con los patrones observados. La publicidad que puede aparecer en cualquier aplicación tal como *YouTube*, o en redes sociales como *Facebook*, *Instagram*, *Twitter* o *TikTok*, se encuentra personalizada de acuerdo a los patrones de la persona, los sitios que visita, los artículos que compra o las páginas de famosos y empresas que sigue. Detrás de toda esta publicidad hecha a la medida para el usuario, hay algoritmos de ML en constante adaptación a los nuevos datos emitidos con cada clic, con cada *Me gusta* o con cada *retweet*.

Cuando una aplicación de correo electrónico filtra los *e-mails* recibidos de acuerdo a si son spam o no, se está haciendo uso de un algoritmo de ML. Los buscadores de internet se apoyan del ML para entender lo que el usuario realmente desea encontrar al escribir en la barra de búsqueda, además de que permiten entender las palabras introducidas, a pesar de estar mal escritas. Otro ejemplo muy popular es el de la computadora "Deep Blue" que fue capaz de superar por primera vez al entonces campeón mundial de ajedrez en 1997. Incluso los populares autos de Tesla que presumen de capacidades de manejo automático tienen como base de funcionamiento este tipo de algoritmos. Otra aplicación presente en la vida diaria y que puede llegar a pasar desapercibida es la detección de compras fraudulentas con tarjetas de crédito, donde el algoritmo es capaz de detectar si una compra está fuera de los hábitos de consumo de esa persona.

Son tres los puntos clave para el éxito generalizado que ha tenido el ML y su gran popularidad como alternativa de solución frente a métodos tradicionales, estos puntos son: mantenimiento, mitigación de riesgos y ventajas (Liu, 2020). Como es bien sabido, la implementación de máquinas y computadoras de manera generalizada en la sociedad actual se ha dado, en parte porque éstas no necesitan dormir, pueden cumplir sus tareas las veinticuatro horas del día, los siete días de la semana, aunque exigen mantenimiento y actualización, a largo plazo representan una reducción de costos.

Tradicionalmente, las computadoras necesitan que se les defina una serie de reglas extensas y órdenes para ejecutar una tarea, es más conveniente y eficiente desarrollar un algoritmo de ML que pueda aprender y extraer patrones de los datos para resolver un problema ya que no requiere

mantenimiento o actualización en sus reglas como un programa de computadora tradicional debido a que, en estos algoritmos se pretende simular el razonamiento que tendría un ser humano.

Por otra parte, las máquinas son más precisas y rápidas, permiten resolver en una fracción de tiempo mucho menor distintas tareas que a un humano le costarían mucho más tiempo y recursos, sin mencionar que estas tareas pueden ser increíblemente repetitivas y tediosas. El ML permite mitigar riesgos latentes ya que puede realizar la misma tarea un millón de veces sin errores que serían esperados si la tarea la realizara un humano.

Finalmente, es mucho más sencillo y barato implementar algoritmos que puedan "aprender" a realizar una tarea, que preparar a un ser humano para que se vuelva experto en ella. Por sí solos, los algoritmos de ML no son suficientes para superar a un grupo de humanos, sin embargo, un individuo preparado y la ayuda de algoritmos de ML sí puede superar a un grupo de expertos (Liu, 2020).

1.1.1. Aplicaciones del *Machine Learning* en la ingeniería petrolera

El ML es un campo que está pasando por una gran expansión en distintas disciplinas, mostrando un desempeño generalmente notable y una capacidad sin precedentes para adaptarse a distintos problemas y contextos. La ingeniería petrolera no es ajena a esto, existen en la literatura diversos antecedentes de aplicación del ML en la industria, desde aplicaciones en el área de perforación de pozos, pasando por producción y caracterización de yacimientos hasta aplicaciones en recursos no convencionales.

Para dar algunos ejemplos de las aplicaciones que se están estudiando sobre el ML en temas relacionados a la ingeniería petrolera se tiene el trabajo de Odi y Nguyen, 2018, destinado a la predicción de facies geológicas, Odutola y col., 2022, analizan su aplicación en gestión de riesgos por formación de hidratos, Bandura y col., 2018, exploran el uso en interpretación sísmica, Jayeola y col., 2022, hacen lo mismo para el análisis de curvas de declinación, Ogwu y col., 2022, lo emplean para la predicción de precios de gas natural, Ugoyah y col., 2022, exploran su uso en aseguramiento de flujo, Mal y col., 2022, analizan la posibilidad de implementación para evitar el atrapamiento de tuberías, Rogulina y col., 2022, aplican estos algoritmos en registros de pozo, Palmer y Gu, 2022, describen su aplicación en terminación de pozos para yacimientos no convencionales; estos trabajos serán retomados en el capítulo tres de una forma más extensa. Todos estos ejemplos solo representan una fracción de las áreas en las cuales el ML está teniendo una incursión, se espera que a medida que gane más popularidad y confianza, sus aplicaciones sean cada vez más variadas y especializadas.

Un área de oportunidad para la aplicación del ML se encuentra en la predicción de propiedades PVT cruciales para la caracterización de yacimientos y consecuentemente, para los pronósticos de producción de un pozo, yacimiento o campo. Tradicionalmente, las propiedades PVT para un fluido hidrocarburo son determinadas mediante pruebas especializadas de laboratorio que requieren todo un procedimiento para obtener resultados, desde estabilizar el pozo para realizar un muestreo y asegurar la captura de fluidos representativos del subsuelo, el transporte de las muestras a un laboratorio especializado. Lo anterior se traduce en una inversión de tiempo y recursos que la empresa encargada debe de cubrir.

Alternativamente se suelen emplear correlaciones empíricas que permiten, aproximar los valores de las propiedades de la mezcla de hidrocarburos. Estas correlaciones empíricas suelen estar limitadas por rangos de aplicabilidad en sus parámetros de entrada, ya que cada correlación es desarrollada con muestras de fluidos de cierta zona y en cierto rango de valores; además de esto, carecen de fundamentos físicos en su desarrollo. Ejemplos de este tipo de aplicaciones y, analizadas en capítulos posteriores son los trabajos de Ramirez y col., 2017, Osman y col., 2001, Yang y col., 2020, Varotsis y col., 1999, Onwuchekwa, 2018 y Numbere y col., 2013.

1.2. Objetivos

De acuerdo con lo explicado en la sección anterior se presentan los objetivos de esta tesis.

Generales

Analizar la capacidad que tiene el ML para la predicción de propiedades PVT en distintos escenarios, al evaluar el desempeño de seis algoritmos de ML cualitativa y cuantitativamente. Desarrollar una herramienta computacional portable y de fácil manejo que implemente algoritmos de ML para predecir propiedades de hidrocarburos que, usualmente son determinadas mediante pruebas de laboratorio PVT o a través de correlaciones empíricas.

Particulares

- Investigar las correlaciones empíricas más utilizadas para analizar el comportamiento de los fluidos hidrocarburos a condiciones de yacimiento.
- Examinar a través de la literatura las aplicaciones relacionadas con el ML y la industria petrolera, específicamente en relación con la predicción de propiedades PVT para fluidos hidrocarburos.
- Desarrollar una herramienta computacional que aplique técnicas de ML con la capacidad de predecir algunas de las propiedades PVT relacionadas con trabajos previos disponibles en la literatura, tomando como datos de entrada para este software variables relativamente fáciles de obtener de fluidos producidos en yacimientos de México y otras regiones del mundo.
- Analizar el desempeño de la herramienta computacional y proponer su implementación para la determinación de propiedades PVT en yacimientos nacionales e internacionales.

1.3. Estructura de la tesis

En este trabajo se presenta un análisis cuantitativo y cualitativo de distintos algoritmos de ML aplicados a la ingeniería petrolera, específicamente a la predicción de propiedades PVT que típicamente son obtenidas mediante un análisis de muestras en laboratorios o por medio de una variedad de correlaciones empíricas aplicables en cierto rango de valores para sus parámetros.

Para hablar de propiedades PVT, primero es necesario describir los distintos tipos de yacimientos, su composición y el comportamiento que estos tienen, es por esto que en el capítulo dos de este trabajo se detallan los tipos de yacimiento de acuerdo a su diagrama de fases, así como el comportamiento de cada uno y el rango de valores típicos en sus propiedades, también se describen las principales propiedades de los fluidos petroleros, tanto de la fase aceite como de la fase gaseosa, así como el comportamiento general de estas propiedades. Posteriormente, se explican los principales métodos de muestreo empleadas en pozos para recuperar muestras de fluidos representativos del yacimiento, subsecuentemente, se describen las pruebas de laboratorio más usuales a las que son sometidas estas muestras con fluidos, se esquematiza el procedimiento de cada test y se menciona la utilidad y propósito de cada uno de estos ensayos. Para cerrar este capítulo se habla de las correlaciones empíricas y su uso para la estimación de propiedades PVT, también se hace mención de las correlaciones más populares y los parámetros que estas pueden predecir.

En el capítulo tres se explican los principales conceptos asociados con el ML, se describen brevemente los principales tipos de aprendizaje reportados en la literatura así como los tipos de algoritmos que existen con base en el problema y tipo de datos, de igual forma se presentan de manera general los datos, su importancia en el ML, su uso y el proceso de filtrado de estos. Posteriormente, se detalla el principio de funcionamiento de algunos algoritmos de ML (algoritmos empleados en capítulos posteriores), se analizan las principales métricas encargadas de evaluar el desempeño de los algoritmos, se examina la diferencia entre parámetros e híper parámetros y finalmente se muestran algunos antecedentes de aplicación de ML a la industria petrolera, especialmente aquellos relacionados con el tema medular de este trabajo.

En el cuarto capítulo se discute detalladamente la metodología de esta tesis, se menciona el lenguaje de programación y paquetería usada, así como la versión de cada una, se describen los distintos modelos de ML que se emplearán, más adelante, se muestra el proceso de adquisición, filtrado y preprocesado de los datos de entrada, de igual forma se explica el uso de correlaciones y de ML para la generación de un set de datos de entrada más completo. Seguido de esto se muestra el set de datos definitivo que se obtuvo. Por último se enumeran los distintos casos donde los algoritmos de ML serán aplicados y las variables a predecir.

El capítulo cinco abarca el análisis de los resultados generados con la aplicación de los algoritmos a los distintos casos. Se enumeran los mejores algoritmos para cada caso así como los parámetros de cada modelo, se muestra un análisis cualitativo y cuantitativo de cada modelo por medio de gráficos de error relativo, absoluto, comparación con los datos reales y con tablas que incluyen la descripción estadística de los datos generados, error medio relativo, absoluto y coeficiente de correlación para cada algoritmo. Además de los modelos, se generaron los mismos datos para algunas correlaciones con el objetivo de comparar el desempeño de estos contra los métodos usuales empleados para estimar propiedades PVT.

El último capítulo habla de las conclusiones derivadas de este trabajo y sus resultados. Se mencionan las posibles aplicaciones que puede tener el ML como herramienta auxiliar para las predicciones de propiedades de hidrocarburos y las implicaciones que esto puede generar. Además, se dan algunas recomendaciones con relación al ML y la predicción de propiedades PVT.

Adicionalmente, se tienen tres apéndices, el primero muestra las ecuaciones pertenecientes a las correlaciones usadas como apoyo en este trabajo, se menciona el rango de valores para los parámetros con que fueron desarrolladas y se explica brevemente de donde provienen las muestras de fluidos que cada autor usó.

Como segundo apéndice, se presenta en pseudocódigo la herramienta computacional que permitió realizar la selección de los mejores algoritmos de ML y su aplicación posterior para generar datos de las variables objetivo, mismos que son ocupados en el capítulo cinco para el análisis de resultados.

En el tercer y último apéndice se muestran las tablas completas con la descripción estadística de los datos generados para los seis algoritmos aplicados a la predicción de la presión de burbuja y el factor de volumen de formación del aceite a la presión de burbuja, así como el error relativo, error absoluto y coeficiente de correlación para cada uno.

Todas las gráficas, tablas y figuras generadas por el autor aparecen sin ninguna leyenda adicional más que la descripción propia que aplique en cada caso.

Capítulo 2

Propiedades de los fluidos petroleros

En este capítulo se describe brevemente la manera de clasificar los yacimientos de hidrocarburos con base en su presión inicial y su diagrama de fases, se mencionan las propiedades más importantes para caracterizar la fase líquida (aceite) y gaseosa, se explican los métodos de muestreo, responsables de recolectar muestras de fluidos que mantengan las condiciones necesarias para ser representativas de los fluidos contenidos en el yacimiento, muestras que, son necesarias para las pruebas de laboratorio PVT. Seguido de esto, se detalla el procedimiento llevado a cabo en las pruebas de laboratorio más usuales, encargadas de determinar las propiedades de una mezcla particular de hidrocarburos. Finalmente, se habla de las correlaciones empíricas y su utilidad como herramienta para predecir propiedades PVT.

2.1. Clasificación de yacimientos

Existe más de una forma de clasificar a los yacimientos petroleros, esto es en función de su diagrama de fases, por el tipo de fluidos presentes, por su presión de yacimiento y de saturación, entre otras. Para este trabajo, resulta particularmente útil clasificar a los yacimientos en función de la presión inicial y la presión de saturación de los fluidos contenidos en éste. Por lo tanto, es necesario entender lo que es un diagrama de fases o diagrama presión-temperatura y definir algunos conceptos para hablar de su clasificación. En la figura 2.1 se muestra un diagrama de fases típico, mientras que en los siguientes párrafos se describen los parámetros más importantes de este diagrama.

Envolvente de fases, delimitada por la curva de puntos de burbuja (A) y la curva de puntos de rocío (B) señaladas en la figura 2.1. Es bajo está región, que se dan las condiciones de presión y temperatura que provocan un equilibrio de fases (gas-líquido) es decir, se tienen presentes dos fases, la proporción en la que cada fase se encuentra está indicada por las líneas de iso-volumen.

Líneas de isovolumen, son las líneas dentro de la envolvente de fases que presentan un volumen igual de líquidos.

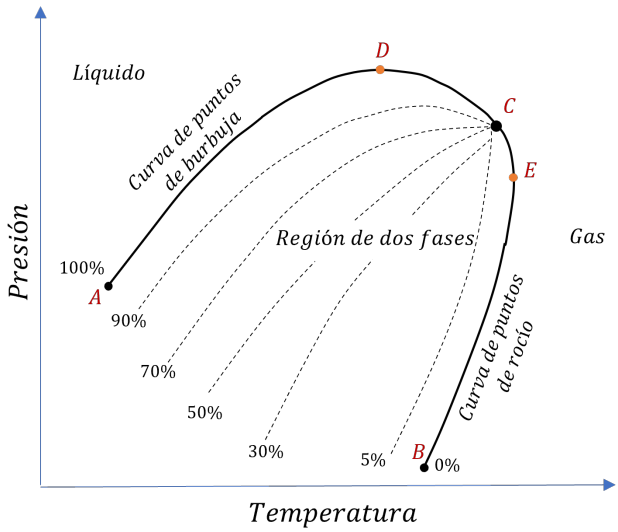


Figura 2.1: Diagrama de fases típico, modificado de Ahmed, 2018, p.2

Curva de puntos de burbuja, es la curva formada por los puntos a los cuales las condiciones de presión y temperatura forman trazas de gas. Delimita a la envolvente de fases, en la figura 2.1 está señalada con la letra "A".

Curva de puntos de rocío, es la curva formada por los puntos a los cuales las condiciones de presión y temperatura forman trazas de líquido. Delimita a la envolvente de fases, en la figura 2.1 está señalada con la letra "B".

Punto crítico, en este punto las condiciones de presión y temperatura provocan que las propiedades intensivas de la fase gas y líquida sean indistinguibles una de otra. Este punto corresponde a la presión crítica y la temperatura crítica del fluido, señalado con la letra "C" en la figura 2.1.

Cricondenbara, es la máxima presión a la que pueden coexistir líquido y gas. Esta presión se encuentra al nivel señalado por la letra "D" en el diagrama de la figura 2.1.

Cricondenterma, se refiere a la máxima temperatura a la que pueden coexistir líquido y gas. La letra "E" representa a la cricondenterma en la figura 2.1.

De acuerdo con Ahmed, 2018 es posible clasificar a los yacimientos en distintas categorías con base en su diagrama de fases, es importante mencionar que esta clasificación es ampliamente usada en la industria debido a su utilidad, aunque esta es una clasificación general y un yacimiento particular puede no entrar solamente en una categoría.

Además de esto, los yacimientos de aceite pueden dividirse en dos categorías con base en su presión, cuando un yacimiento de aceite tiene una sola fase, es decir, que su presión se encuentra

por encima de la presión de burbuja, se dice que es un yacimiento *bajosaturado*, de manera análoga, cuando la presión del yacimiento es menor a la presión de burbuja, se le denomina yacimiento *saturado*.

2.1.1. Yacimientos de aceite

A continuación se presenta la clasificación de yacimientos de aceite propuesta por McCain, 1990.

Yacimiento de aceite negro

De acuerdo con McCain, 1990, se caracterizan por tener una curva de puntos de burbuja (A) en una posición casi horizontal y líneas de isovolumen prácticamente igual de espaciadas, un ejemplo de este tipo de yacimientos se muestra en la figura 2.2. Por otro lado, la presión del yacimiento suele encontrarse por encima de la curva de puntos de burbuja y tiende a entrar a la envolvente a medida que decae la presión (puntos 1 y 2) y las condiciones superficiales suelen caer en el área de dos fases (punto 3). Estos yacimientos tienen densidades API y relaciones gas-aceite muy pequeñas, menores a 45 grados y 2,000 SCF/STB respectivamente.

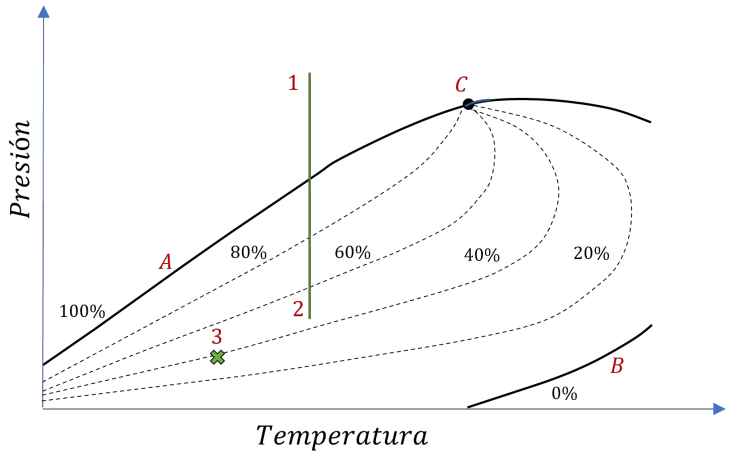


Figura 2.2: Diagrama de fases típico para aceite negro, modificado de McCain, 1990, p.150

Yacimiento de aceite volátil

También llamados de alto encogimiento, de acuerdo a McCain, 1990, tienen relativamente menos moléculas de componentes pesados y una mayor proporción de componentes intermedios (C_2-C_6), como puede verse en la figura 2.3, tienen una curva de puntos de rocío más pronunciada (B). Las líneas de isovolumen se encuentran más espaciadas hacia la curva de puntos de rocío mientras que son mas estrechas cerca de la curva de puntos de burbuja (A). Las condiciones iniciales del yacimiento se encuentran más cerca del punto crítico y a medida que entra en producción, se desarrolla una mezcla bifásica en yacimiento (puntos 1 y 2), en superficie se tiene típicamente una mezcla de dos fases

(punto 3). Este tipo de yacimientos presentan líquidos con un alto encogimiento en superficie. La densidad API suele ser mayor a 40 grados y presentan relaciones gas-aceite de 2,000 a 3,200 SCF/STB.

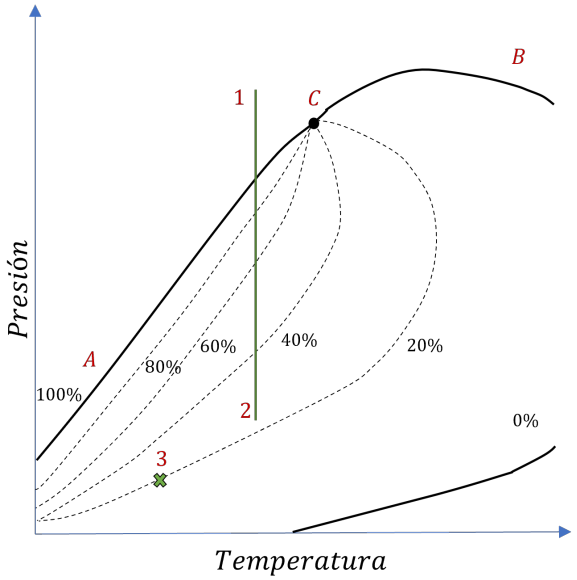


Figura 2.3: Diagrama de fases típico para aceite volátil,
modificado de McCain, 1990, p.152

2.1.2. Yacimientos de gas

En esta sección se muestran los distintos tipos de yacimientos de gas de acuerdo a la clasificación tomada por McCain, 1990.

Yacimiento de gas y condensado

De acuerdo con McCain, 1990, son yacimientos en los cuales la temperatura del yacimiento se encuentra entre la cricondenterma y la temperatura crítica. Como se puede observar en la figura 2.4, la envolvente de fases tiene una forma más vertical y la curva de puntos de rocío (B) es más pronunciada que la curva de puntos de burbuja (A), las condiciones iniciales provocan que se tenga solamente gas en el yacimiento, sin embargo, al entrar a producción, la presión del yacimiento entra en la envolvente de fases por lo que se tiene una mezcla de gas y condensados en el yacimiento, al disminuir aún más la presión, debido a la forma de su diagrama presión-temperatura se evapora parte del fluido líquido y aumenta la producción de gas (puntos 1 y 2), por lo que las moléculas de hidrocarburos más pesadas se queden en el yacimiento. Suele tenerse una mezcla bifásica en superficie (punto 3). Se tienen relaciones gas aceite iniciales de 3,300 SCF/STB, esta proporción tiende a subir hasta aproximadamente 150,000 SCF/STB ya que los componentes más pesados quedan atrapados en el yacimiento y la densidad API suele ir de los 40 a los 60 grados.

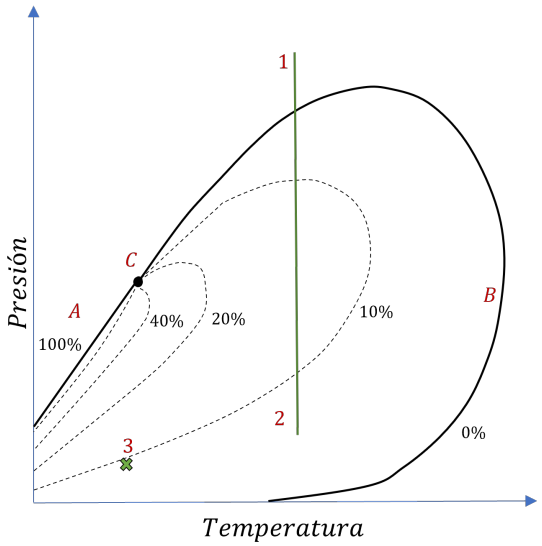


Figura 2.4: Diagrama de fases típico para gas y condensado retrógrado, modificado de McCain, 1990, p.154

Yacimiento de gas húmedo

De acuerdo con McCain, 1990, la temperatura inicial del yacimiento se encuentra por encima de la cricondenterma, debido a esto, solamente a lo largo de toda la vida productiva del yacimiento solamente se tendrá gas en el subsuelo, este comportamiento puede ser observado en los puntos 1 y 2 de la figura 2.5, sin embargo, las condiciones superficiales se encuentran dentro de la envolvente por lo que se tendrá una mezcla bifásica en el separador (punto 3). Este tipo de yacimientos tienen una RGA mayor a 50,000 SCF/STB y los condensados producidos mantienen una gravedad API constante a lo largo de la vida del yacimiento.

Yacimiento de gas seco

De acuerdo con McCain, 1990, este tipo de yacimiento al igual que el de gas húmedo, tiene una temperatura inicial mayor a la cricondenterma por lo que siempre se tiene una sola fase gaseosa en la formación. En superficie también se tiene una sola fase, es decir, nunca se entra a la región bifásica (puntos 1, 2 y 3 en la figura 2.6), además la curva de puntos de rocío es bastante pronunciada (B).

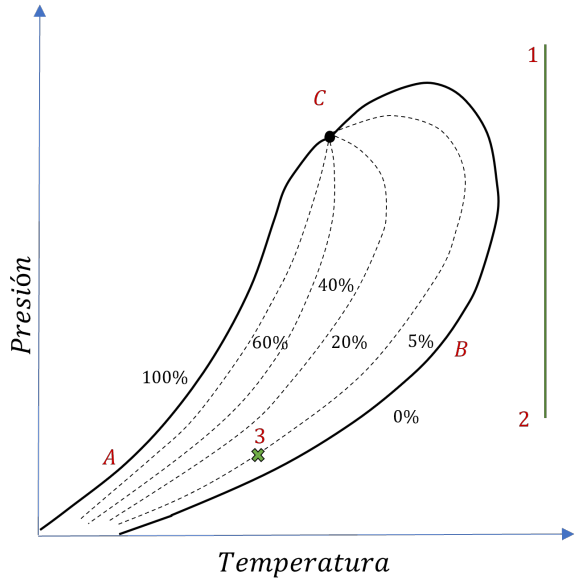


Figura 2.5: Diagrama de fases típico para gas húmedo, modificado de McCain, 1990, p.157

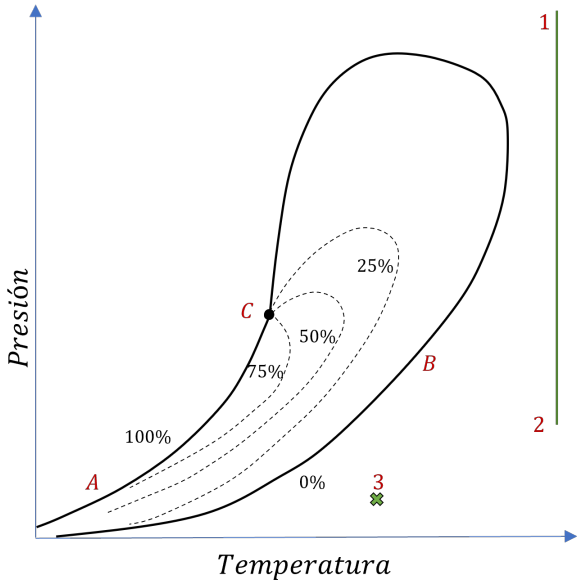


Figura 2.6: Diagrama de fases típico para gas seco, modificado de Ahmed, 2018, p.15

2.2. Propiedades de los fluidos petroleros

La mezcla de hidrocarburos contenidos en un yacimiento tienen una composición particular, con esto en mente, es de esperarse que las propiedades que dictan su comportamiento sean específicas para cada mezcla.

Estas propiedades son críticas para determinar el comportamiento volumétrico del yacimiento. Generalmente para obtener estas propiedades es necesario tomar muestras y realizar estudios de laboratorio PVT o usar correlaciones empíricas, a continuación, se resumen las propiedades de los fluidos más importantes para determinar el comportamiento del yacimiento.

2.2.1. Propiedades del aceite

Factor de volumen de formación, B_o

Está definido como el volumen de aceite de yacimiento (incluyendo gas disuelto) que se necesita para obtener un barril de aceite a condiciones estándar (véase ecuación 2.1). En la figura 2.7 se muestra el comportamiento de esta propiedad respecto a la presión, se observa que por encima de la presión de burbuja el factor de volumen de formación tiene un comportamiento aproximadamente constante, mientras que por debajo de la presión de burbuja la pendiente en la gráfica aumenta abruptamente debido a la liberación de componentes ligeros.

$$B_o = \frac{\left(V_o + V_{gd}\right)_{p,T}}{\left(V_o\right)_{ce}} \tag{2.1}$$

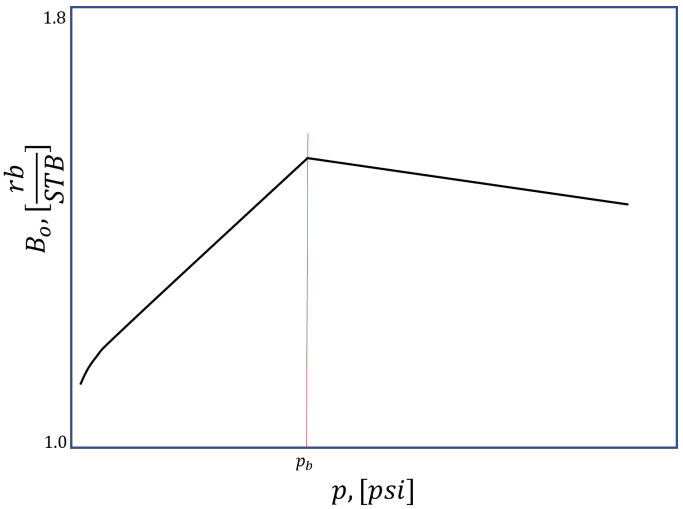


Figura 2.7: Comportamiento típico del factor de volumen de formación del aceite en función de la presión, modificado de McCain, 1990, p. 227

Densidad relativa del aceite, γ_o

Es el cociente de la densidad del aceite y la densidad del agua, ambas a condiciones específicas de presión y temperatura; también conocida como gravedad específica (véase ecuación 2.2).

$$\gamma_o = \frac{(\rho_o)_{p,T}}{(\rho_w)_{p,T}} \tag{2.2}$$

Densidad API, $^\circ API$

En la industria petrolera se acostumbra expresar la densidad relativa en grados API, la relación se detalla en la ecuación 2.3, su relación con la densidad relativa es inversa, a mayor densidad relativa, menor densidad API y viceversa.

$$^\circ API = \frac{141.5}{\gamma_o} - 131.5 \tag{2.3}$$

Relación de solubilidad, R_s

Este parámetro mide el volumen de gas disuelto presente en el aceite a condiciones de yacimiento, tanto el volumen de gas disuelto como el del aceite son medidos a condiciones superficiales, en figura 2.8 se puede ver el comportamiento de esta propiedad. Por encima de la presión de burbuja esta magnitud se mantiene constante, una vez que la presión decae y comienzan a liberarse componentes gaseosos, la relación de solubilidad disminuye gradualmente (véase ecuación 2.4).

$$R_s = \frac{\left((V_g)_{p,T}\right)_{ce}}{(V_o)_{ce}} \tag{2.4}$$

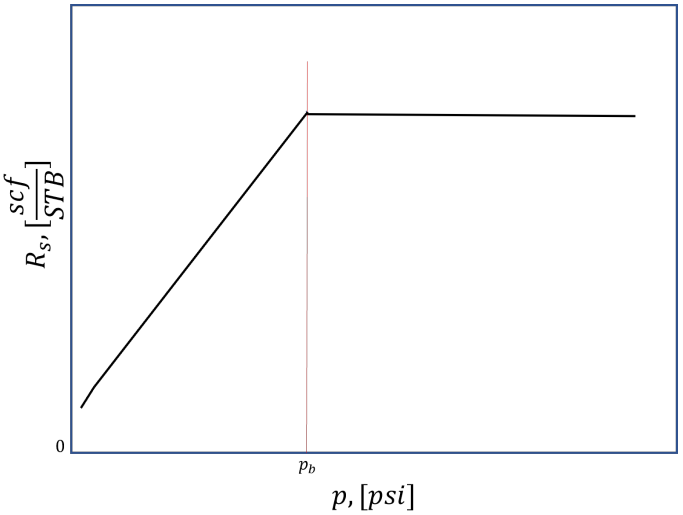


Figura 2.8: Comportamiento típico de la relación de solubilidad en función de la presión, modificado de McCain, 1990, p.228

Relación gas-aceite, RGA

Este parámetro de producción se incluye en esta sección ya que, será de utilidad para los siguientes capítulos. Se define como la relación del volumen de gas producido por el volumen de aceite producido, ambos registrados a condiciones superficiales. La figura 2.9 muestra el comportamiento de esta variable en función de la presión. Como se mencionó previamente, conforme se entra a la envolvente de fases, la proporción de gas aumenta en el yacimiento, es decir, comienzan a liberarse en forma de gas las moléculas más ligeras, por lo que a medida que la presión declina cada vez más, la RGA aumenta gradualmente.

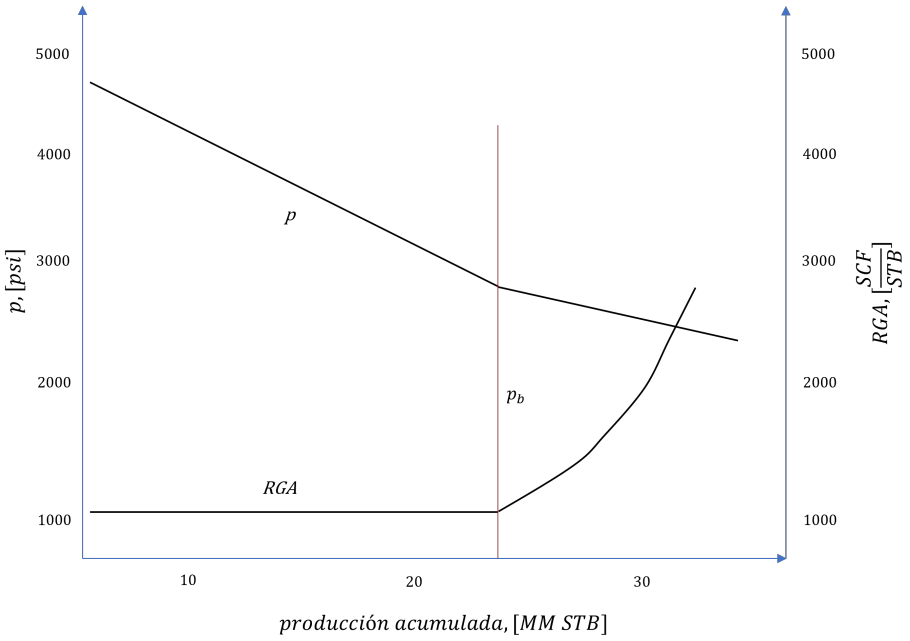


Figura 2.9: Comportamiento típico de la relación gas-aceite en función de la presión, modificado de McCain, 1990, p.250

Coeficiente de compresibilidad isotérmica del aceite, C_o

Como puede verse en la figura 2.10, representa el cambio fraccional en volumen del aceite en función de la presión, manteniendo la temperatura constante. El valor de esta magnitud aumenta a medida que la presión decae; es importante mencionar que el comportamiento mostrado en esta figura solamente aplica para cuando se yacimientos con presión inicial mayor a la presión de burbuja (véase 2.5).

$$c_o = -\frac{1}{V}\left(\frac{\partial V}{\partial p}\right)_T \tag{2.5}$$

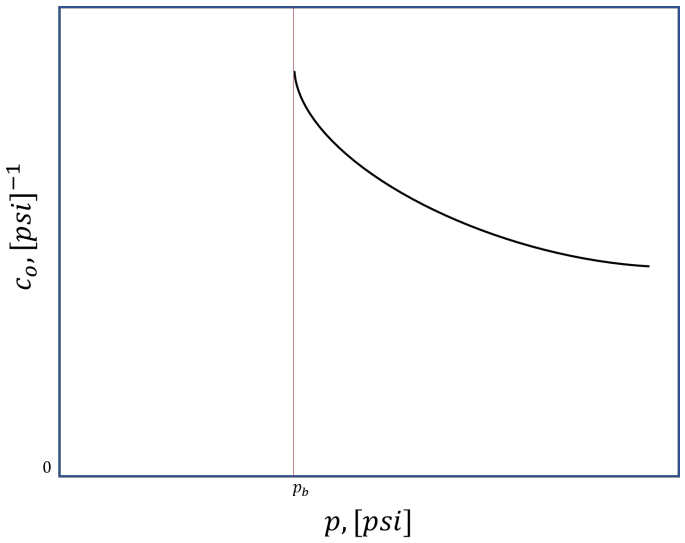


Figura 2.10: Comportamiento típico del coeficiente de compresibilidad isotérmica del aceite en función de la presión, modificado de McCain, 1990, p.232

Viscosidad del aceite, μ_o

Es la medida de la resistencia al esfuerzo de corte ejercida sobre el aceite. Su valor es función de la presión y temperatura, a medida que la temperatura aumenta, la viscosidad decrece, por otro lado, si la presión aumenta, la viscosidad también aumentará (solamente cuando la presión es mayor a la presión de burbuja). Cuando la presión es menor a la presión de burbuja la viscosidad tiene un comportamiento opuesto y vuelve a aumentar a medida que sigue declinando la presión, esto debido a que el líquido comienza a liberar los componentes más ligeros (véase figura 2.11).

Presión de burbuja, p_b

La presión de burbuja es un parámetro esencial para la caracterización de un yacimiento, está referida como la presión a la cual el aceite libera la primera burbuja de gas. Como se puede ver en las figuras 2.7, 2.8, 2.10, 2.11 y 2.12 que, la tendencia de las propiedades cambia abruptamente cuando se alcanza la presión de burbuja.

Factor de volumen de formación total, B_t

Se define como el cambio total en volumen que experimenta el fluido desde el yacimiento hasta superficie, incluye al aceite, al gas disuelto en el aceite y al gas libre (véase figura 2.12), por encima de la presión de burbuja este parámetro se mantiene prácticamente constante, sin embargo, una vez que se manifiestan las dos fases en el yacimiento, el valor de B_t aumenta gradualmente debido a la rápida expansión de la fase gaseosa (véase ecuación 2.6).

$$B_t = B_o + B_g(R_{sb} - R_s) \tag{2.6}$$

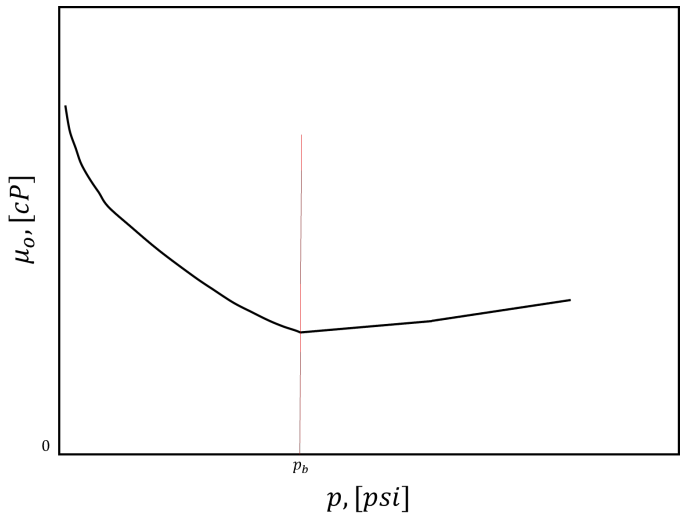


Figura 2.11: Comportamiento típico de la viscosidad del aceite en función de la presión a temperatura constante, modificado de McCain, 1990, p.237

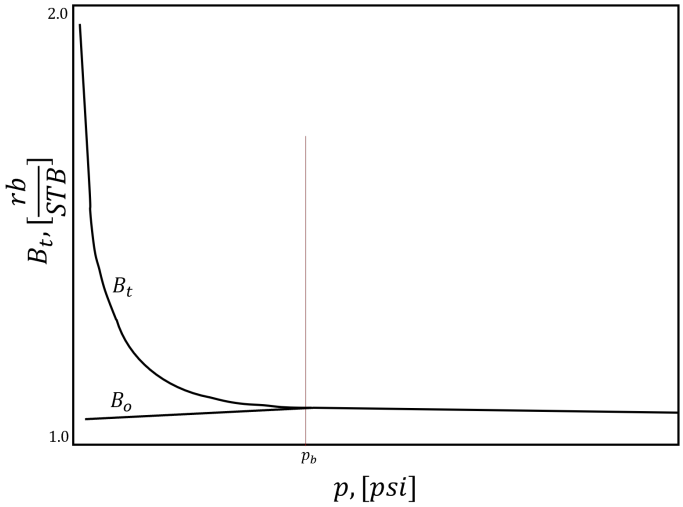


Figura 2.12: Comportamiento típico del factor de volumen de formación total en función de la presión, modificado de McCain, 1990, p.230

2.2.2. Propiedades del gas

El gas en un yacimiento petrolero usualmente es una mezcla de hidrocarburos ligeros y gases no hidrocarburos o contaminantes, es común encontrar componentes como metano (C_1), etano (C_2), propano ($C3$) y butano (C_4) en grandes cantidades mientras que componentes más pesados se presentan en cantidades reducidas, por otro lado, los gases contaminantes más comunes son

dióxido de carbono (CO_2), nitrógeno (N_2) y ácido sulfhídrico (H_2S). A continuación se resumen las principales propiedades de la fase gas.

Masa molecular aparente del gas, M_a

Está definido como la suma del producto de la masa molecular de cada componente multiplicado por la fracción mol que ocupa ese componente en la mezcla (véase ecuación 2.7).

$$M_a = \sum_{i=1}^{n} y_i M_i \tag{2.7}$$

Factor de desviación del gas, z

Es un factor de ajuste introducido a la ley de los gases ideales para representar adecuadamente el comportamiento de un gas real. También llamado factor de compresibilidad; es el cociente del volumen de gas con un comportamiento real dividido entre el volumen que ocuparía si tuviera un comportamiento ideal (véase ecuación 2.8).

$$z = \frac{V_{real}}{V_{ideal}} \tag{2.8}$$

Densidad del gas, ρ_g

Está definida a través de la ecuación de compresibilidad de los gases, se obtiene de la siguiente:

$$\rho_g = \frac{pM_a}{RTz} \tag{2.9}$$

Densidad relativa del gas, γ_g

Es el cociente de la densidad del gas y la densidad de un fluido de referencia (aire), ambos medidos a las mismas condiciones de presión y temperatura (véase ecuación 2.10).

$$\gamma_g = \frac{(\rho_g)_{p,T}}{(\rho_{aire})_{p,T}} = \frac{(M_g)_p, T}{(M_{aire})_p, T} \tag{2.10}$$

Factor de volumen de formación del gas, B_g

Está definido como el volumen de gas a condiciones de yacimiento (o condiciones específicas de presión y temperatura) dividido entre el volumen de la misma cantidad de gas a condiciones estándar (véase ecuación 2.11 y 2.12). Su comportamiento en función de la presión se puede ver en la figura 2.13, donde se puede observar que el factor de volumen de formación del gas aumenta rápidamente a medida que disminuye la presión.

$$B_g = \frac{(V_g)_{p,T}}{(V_g)_{ce}} \tag{2.11}$$

Alternativamente se puede calcular como:

$$B_g = \left(\frac{p}{T}\right)_{ce} \left(\frac{Tz}{p}\right)_{p,T} \tag{2.12}$$

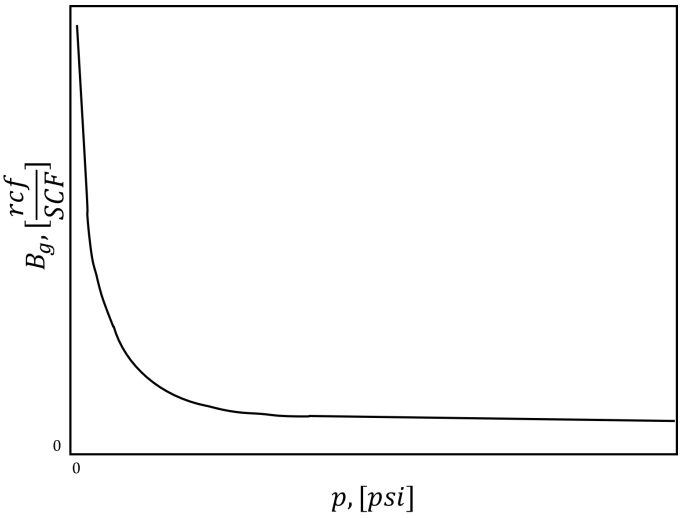


Figura 2.13: Comportamiento típico del factor de volumen de formación del gas en función de la presión, modificado de McCain, 1990, p.168

Coeficiente de compresibilidad isotérmica del gas C_g

Ilustrado en la figura 2.14 y ecuación 2.13, es el cambio fraccional en el volumen del gas en función de la presión a temperatura constante, tiene un comportamiento similar al factor de volumen de formación del gas.

$$c_g = \frac{1}{p} - \frac{1}{z}\left(\frac{\partial z}{\partial p}\right)_T \tag{2.13}$$

Viscosidad del gas, μ_g

Es la medida a la resistencia de un gas a fluir, este valor es significativamente menor a la viscosidad de un líquido debido al mayor espaciamiento entre las moléculas en un gas. De acuerdo con McCain, 1990, la viscosidad de un gas disminuye a medida que la presión decae ya que las moléculas del gas se separan cada vez más por lo que se genera menor fricción entre éstas. Esta propiedad es difícil de obtener experimentalmente, para su determinación suelen emplearse correlaciones empíricas. En la figura 2.15, se muestra el comportamiento de la viscosidad de un gas en función de la presión.

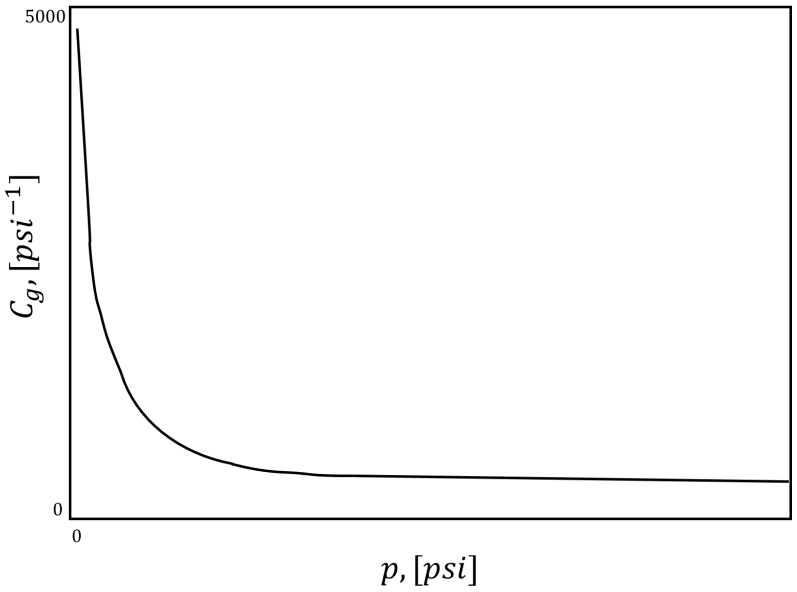


Figura 2.14: Comportamiento típico del coeficiente de compresibilidad isotérmica del gas en función de la presión, modificado de McCain, 1990, p.170

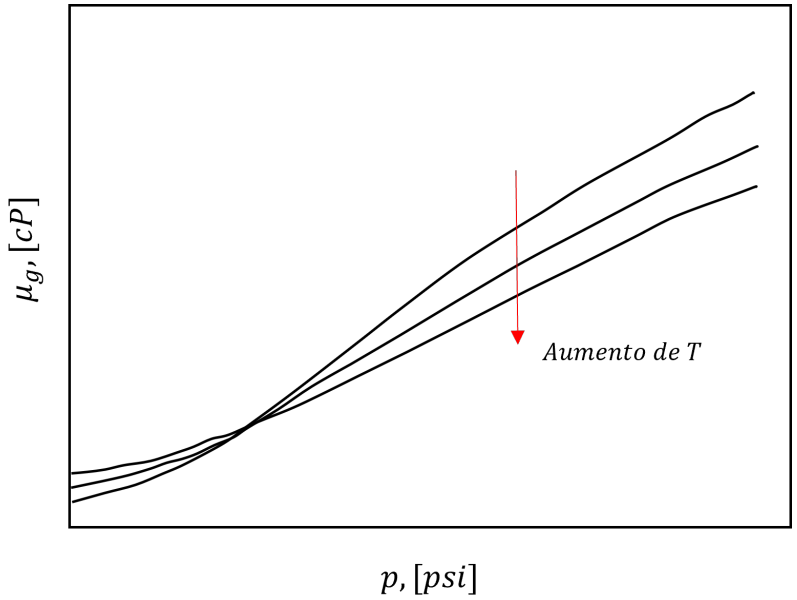


Figura 2.15: Comportamiento típico de la viscosidad del gas en función de la presión, modificado de McCain, 1990, p.179

2.3. Análisis PVT

Para determinar las propiedades de los fluidos contenidos en un yacimiento es necesario realizar un análisis mediante experimentos PVT, estas son diseñadas para conocer las propiedades descritas en la sección anterior y necesarias para caracterizar adecuadamente al yacimiento. Además, se pueden utilizar estos parámetros para generar modelos que permitan predecir el comportamiento futuro del yacimiento y evaluar las estrategias más óptimas para su explotación.

2.3.1. Muestreo de fluidos

Para realizar estas pruebas PVT primero es necesario adquirir una muestra representativa de los fluidos contenidos en el yacimiento. Este procedimiento es conocido como muestreo. Existen varios métodos de muestreo de fluidos, la elección del método más adecuado se determina en función del propósito que se le va a dar a la muestra, el volumen de fluidos requerido, el equipo en superficie disponible, las condiciones mecánicas del pozo, el tiempo de vida del yacimiento, entre otras (American Petroleum Institute, 2003).

Muestreo de fondo de pozo

En este tipo de muestreo, se baja una celda a través del pozo hasta la profundidad a la cual se desea adquirir una muestra de fluidos. Una vez que la celda recolecta el fluido del fondo, es sellada a cierta presión y transportada a superficie. Este método es muy versátil ya que permite adquirir muestras en pozos con agujero descubierto o entubado, su uso está recomendado para pozos donde la presión de fondo fluyente es mayor a la presión de saturación del yacimiento, para evitar la separación en dos fases del fluido a recolectar (American Petroleum Institute, 2003). La figura 2.16 muestra un esquema de un equipo de muestreo con capacidad para mantener las condiciones de presión de fondo, cuenta con un compartimento para almacenar los fluidos de formación y un mecanismo con agua y glicol que es accionado con la ayuda del nitrógeno para asegurar los fluidos, manteniendo las condiciones deseadas (Schlumberger, s.f.-a).

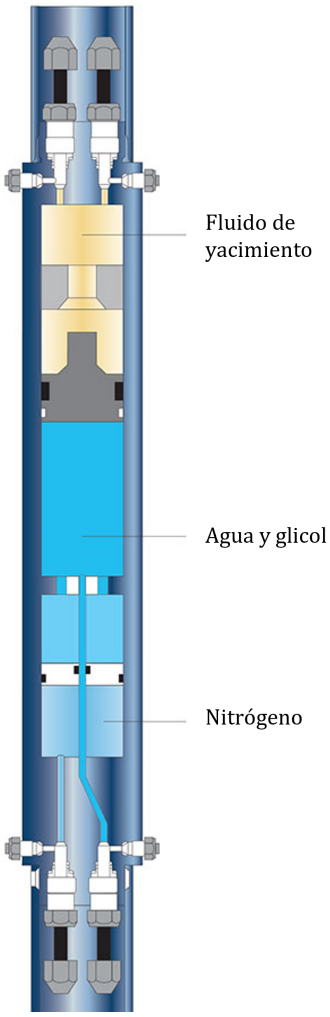


Figura 2.16: Ejemplo de un equipo para muestreo de fondo de pozo, modificado de Schlumberger, s.f.-a

Probadores de formación

Consisten en muestreadores para agujeros descubiertos, al igual que los anteriores, recolectan una muestra del fluido en el fondo de pozo, estos son bajados mediante línea de acero, el dispositivo cuenta con varias celdas que pueden activarse desde la superficie para recolectar las muestras de fluidos directamente de la formación, a diferencia de las celdas de fondo de pozo que recolectan los fluidos que entran al pozo (American Petroleum Institute, 2003).

Muestreo en superficie

Como el nombre lo indica, las muestras son adquiridas en superficie, se toman directamente del flujo que llega al separador, para esto se determinan cuidadosamente los gastos de aceite y de gas que entran en el separador, lo que permitirá recombinar los fluidos de aceite y gas en las proporciones adecuadas una vez se encuentren en el laboratorio.

La ventaja de este método es que es mucho más sencillo y rápido de realizar, además de que permite recolectar grandes cantidades de fluidos, sin embargo, es altamente sensible a la calidad de la medición de gasto de ambos fluidos, cualquier desviación respecto a la proporción real de cada fluido afectará la fidelidad de los resultados de laboratorio ya que el fluido recombinado no será representativo del yacimiento (American Petroleum Institute, 2003).

Muestreo en cabeza de pozo

Figura 2.17: Ejemplo de un equipo para muestreo en cabeza de pozo, tomado de Schlumberger, s.f.-b

Suele usarse cuando se sabe que el fluido a condiciones de presión y temperatura de cabeza de pozo se encuentra en una sola fase. Tiende a aplicarse en yacimientos con condiciones muy específicas, por ejemplo, en aceites bajosaturados o yacimientos de gas seco. Al igual que el muestreo en separador, este es bastante práctico y sencillo de implementar (American Petroleum Institute, 2003). La figura 2.17 muestra un dispositivo móvil que permite la toma de muestras de fluidos en cabeza de pozo a alta presión, consta de seis celdas equipadas con los mecanismos necesarios para atrapar y sellar los fluidos una vez dentro, todas las celdas comparten un tanque de drenado (Schlumberger, s.f.-b).

2.3.2. Pruebas PVT

De acuerdo con Ahmed, 2018, las pruebas pueden dividirse en tres categorías:

1. *Pruebas primarias*: estas consisten en pruebas sencillas realizadas en sitio para determinar la densidad relativa de los fluidos y la relación gas aceite.

2. *Pruebas rutinarias de laboratorio*: son las pruebas más comunes que se realizan sobre los fluidos con el propósito de determinar sus principales propiedades.

3. *Pruebas de laboratorio especiales*: este tipo de pruebas se realizan solo en casos específicos donde se pretenda aplicar una técnica de recuperación secundaria o mejorada en el yacimiento

Las pruebas PVT son realizadas en laboratorios especializados, se realizan con el propósito de obtener los parámetros necesarios para caracterizar al fluido, determinar la mejor estrategia de explotación y realizar pronósticos sobre el comportamiento del yacimiento a futuro.

Pruebas rutinarias

Análisis composicional

En este análisis se determina la composición del fluido, si bien es imposible identificar cada componente en las mezclas de hidrocarburos porque suelen contener miles de componentes diferentes, los componentes más ligeros y sus proporciones pueden determinarse con relativa facilidad, aquellos componentes más pesados suelen agruparse en un solo pseudocomponente (McCain, 1990).

Típicamente se analizaban los componentes hasta el C_6, aunque la constante mejora en la tecnología permite realizar análisis que fácilmente superan la fracción del C_{30}; además de que, las ecuaciones de estado más sofisticadas han demostrado que un análisis más detallado de los componentes arroja resultados más precisos (McCain, 1990).

Esta prueba da como resultado la composición del fluido, con las fracciones molares de cada componente, además de la masa molecular aparente del pseudocomponente más pesado. La figura 2.18 muestra un ejemplo del resultado de esta prueba para una mezcla de gases, las señales en la gráfica pueden identificarse y asociarse con un componente específico.

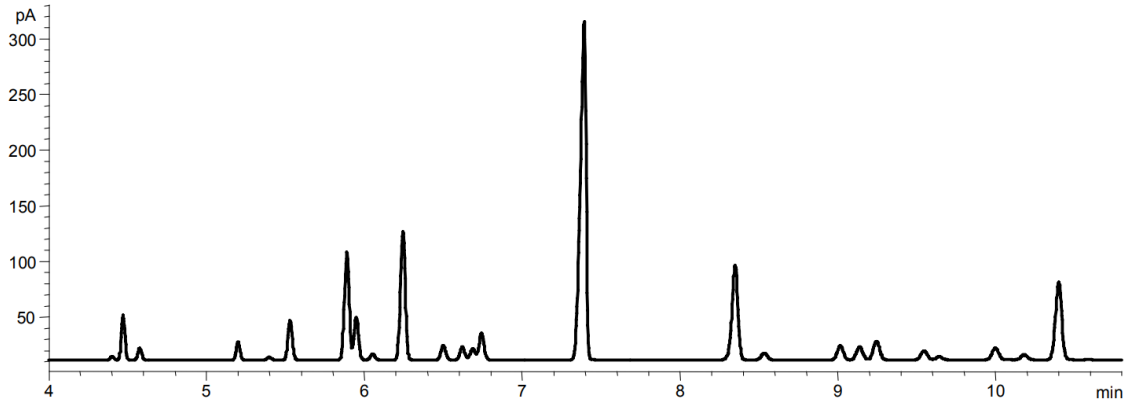


Figura 2.18: Ejemplo del resultado de cromatografía de gases, tomado de Canipa y col., 2003

Separación flash

Esta prueba, también conocida como expansión a composición constante se encuentra esquematizada en la figura 2.19, consiste en llevar la celda, saturada de fluido a una presión igual o mayor que la presión inicial de yacimiento, se realiza con la celda sellada, por lo tanto la composición inicial se preserva durante toda la prueba. Además de esto, la temperatura de la celda se mantiene a la temperatura de yacimiento desde el comienzo hasta el fin de la prueba (McCain, 1990).

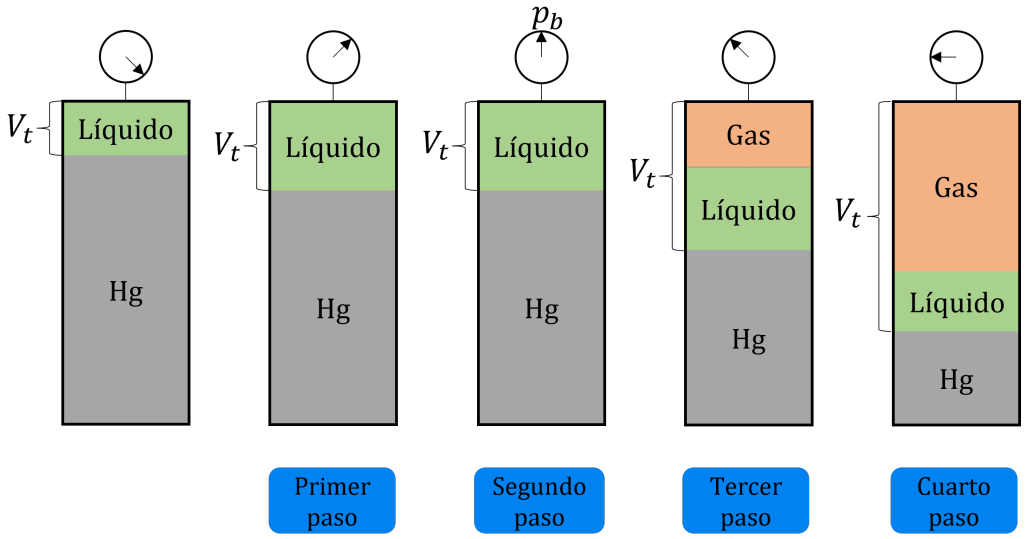


Figura 2.19: Fases de la prueba de separación flash, modificado de McCain, 1990, p.271

La celda se lleva a una presión igual o mayor a la del yacimiento, posteriormente se reduce esta presión por intervalos. En cada uno de estos intervalos se registra el volumen que ocupa el fluido al expandirse, también se registra el volumen relativo.

Esta prueba permite obtener la presión de saturación del fluido, la densidad de éste y el coeficiente de compresibilidad isotérmica (Ahmed, 2018).

Separación diferencial

En esta prueba, la celda PVT se lleva a la presión de burbuja y temperatura de yacimiento. La presión se reduce, conforme esto sucede, el gas en solución se libera. Cada ciertos intervalos de presión, el gas liberado es removido de la celda mientras se mantiene la presión constante, dejando únicamente al líquido. Este proceso es repetido varias veces hasta llegar a condiciones atmosféricas, una vez llegado a este punto, se lleva la temperatura a condiciones estándar (véase figura 2.20). El volumen de gas es medido en cada paso a condiciones estándar, el volumen de aceite residual en cada paso también es registrado (McCain, 1990).

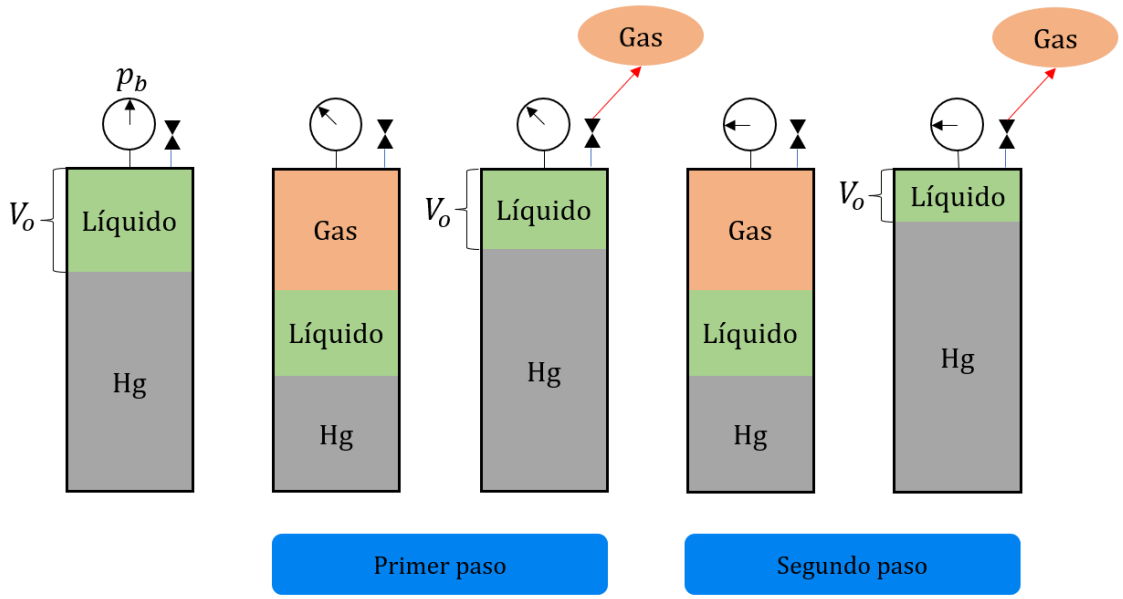


Figura 2.20: Fases de la prueba de separación diferencial, modificado de McCain, 1990, p.274

A diferencia de la prueba anterior, la composición no se mantiene constante ya que en cada intervalo los componentes más ligeros que se liberan, son extraídos, dejando en la celda los componentes más pesados. Esta prueba simula el agotamiento de un yacimiento ya que conforme el gas se libera, este es el el fluido que se produce primero debido a su alta movilidad en relación al aceite (Ahmed, 2018).

De esta prueba se obtiene la relación de solubilidad al sumar todo el gas liberado a condiciones estándar desde la presión de burbuja. También se obtiene el factor de volumen de formación del aceite, del gas y total, la densidad del aceite, el factor de desviación (z), y la densidad relativa del gas (Ahmed, 2018).

Pruebas de separador

En esta prueba se determina el comportamiento que el fluido al ser sometido a diferentes condiciones de separador y a condiciones de tanque de almacenamiento, con esto se busca determinar las condiciones idóneas de separación que permitan maximizar el volumen de aceite en tanque de almacenamiento (McCain, 1990).

Como se puede observar en la figura 2.21, la celda se lleva a temperatura de yacimiento y presión de burbuja, posteriormente el líquido de la celda es llevado a condiciones de presión y temperatura de separador previamente seleccionadas y luego es llevado a condiciones estándar o de tanque de almacenamiento, se registra la densidad relativa del gas en separador y tanque de almacenamiento, además de la densidad API del aceite en tanque de almacenamiento y el factor de volumen de formación del separador. Se busca encontrar el parámetro de presión óptimo, que minimiza el factor de volumen del aceite, y maximiza la densidad API (Ahmed, 2018).

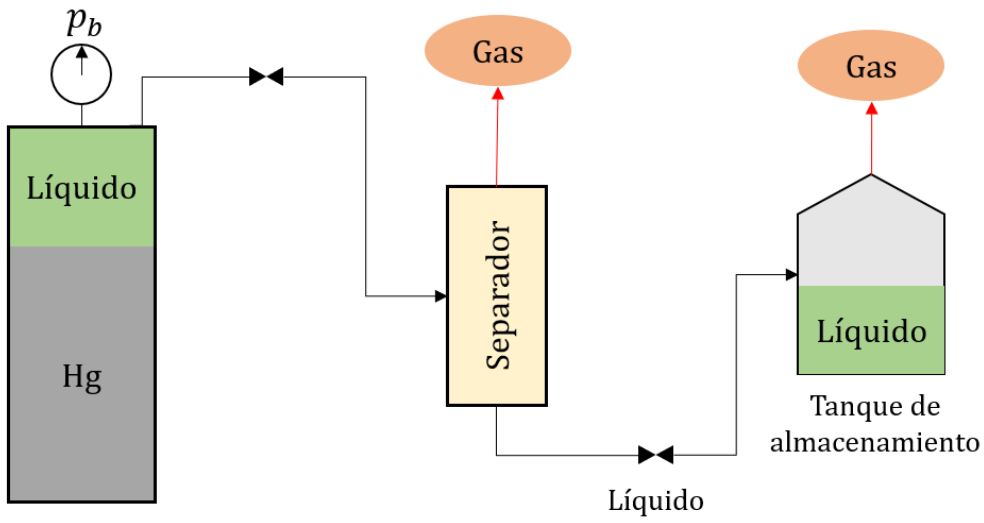


Figura 2.21: Esquema de una prueba de separador, modificado de McCain, 1990, p.277

Prueba de viscosidad

De acuerdo con McCain, 1990, la viscosidad del aceite es determinada en un viscosímetro capilar o rotacional, las mediciones se realizan a diferentes presiones con el aceite residual habiendo retirado el gas liberado en cada paso. Para el gas es usual optar por el uso de correlaciones empíricas para determinar esta magnitud debido a lo complejo que es medirla experimentalmente. Para obtener esta propiedad se toma como base la densidad relativa del gas liberado en la separación diferencial.

Prueba de agotamiento a volumen constante

De acuerdo con Ahmed, 2018, esta prueba se realiza típicamente para gases retrógrados o aceites muy volátiles, al igual que la separación diferencial, esta prueba está diseñada para simular la producción del fluido a lo largo de la vida del yacimiento.

Tomando como ejemplo un gas retrógrado, la celda PVT se lleva a temperatura de yacimiento y presión de rocío, a continuación se disminuye la presión con lo que se liberan condensados, posteriormente se lleva la celda a su volumen original manteniendo la presión constante, simultáneamente se retira una cantidad equivalente de gas de la celda.

Este proceso es repetido varias veces, en cada paso se mide la proporción de volumen de condensados liberados y al gas liberado se le determina su composición y volumen. Este procedimiento simula el agotamiento de un yacimiento de gas y condensados donde el gas es producido mientras que los condensados, al tener menos movilidad, son retenidos en el yacimiento. Este proceso se esquematiza en la figura 2.22.

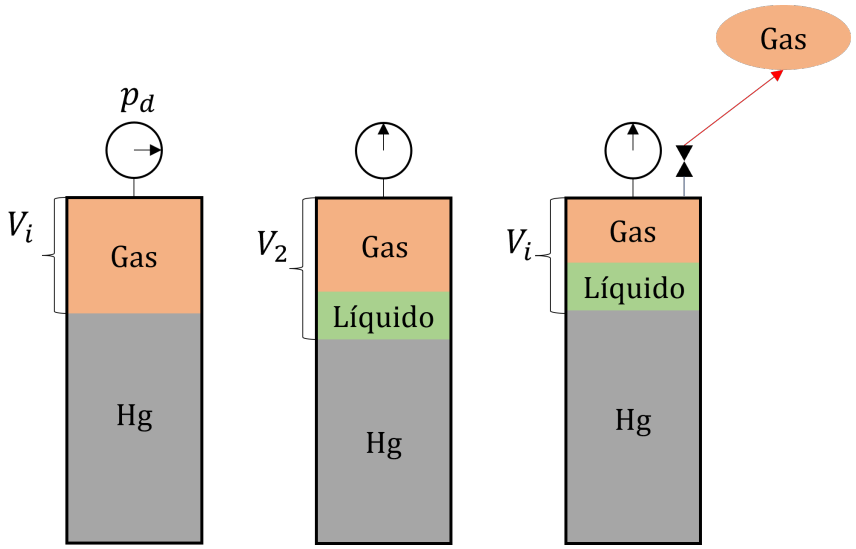


Figura 2.22: Esquema de una prueba de agotamiento a volumen constante, modificado de Ahmed, 2018, p.176

De esta prueba se puede obtener el factor de desviación a la presión de rocío y a cada intervalo de presión, el factor de desviación de dos fases ($z_{2-fases}$) y la relación en porcentaje del volumen de gas producido en función del volumen de gas inicial (Ahmed, 2018).

2.4. Correlaciones empíricas

Muchas veces, los datos de las propiedades PVT de los fluidos del yacimiento no se encuentran disponibles, por lo que es necesario el uso de correlaciones empíricas que permitan obtener valores aproximados de estas propiedades.

Estas correlaciones empíricas son expresiones matemáticas desarrolladas por diversos autores para aproximar el valor de una propiedad con base en los datos más básicos disponibles en campo para un fluido de yacimiento. Se han desarrollado con base en muestras de una cierta región y con ciertos rangos en los datos, por lo que su capacidad predictiva es óptima cuando se aplican en crudos con características similares, por esto, deben emplearse cuidadosamente, tomando en cuenta que son aproximaciones empíricas.

En la tabla 2.1 se muestra una comparación entre algunas de las correlaciones más utilizadas. Incluidas las correlaciones de M. B. Standing, 1977, Glasø, 1980, Al-Marhoun, 1988, Dokla y Osman, 1992 y Petrosky y Farshad, 1993, que serán empleadas más adelante en este trabajo.

Tabla 2.1: Algunas correlaciones empíricas y parámetros que predicen

Correlación	p_b	R_s	B_o	μ_{oD}	μ_o
Standing	✓	✓	✓	-	-
Vázquez & Beggs	✓	✓	✓	-	-
Glasø	✓	✓	✓	✓	-
Al-Marhoun	✓	✓	✓	-	-
Kartoatmodjo & Schmidt	✓	✓	✓	✓	✓
Beggs & Robinson	-	-	-	✓	✓
Beal	-	-	-	✓	-
Dokla & Osman	✓	-	✓	-	-
Perosky & Farshad	✓	✓	✓	-	-

Capítulo 3

Conceptos de *Machine Learning*

En este capítulo se presenta una introducción al *Machine Learning* (ML), se presentan las diferentes corrientes de pensamiento que dan lugar a los algoritmos de ML, así como ciertos conceptos esenciales para adentrarse a este tema. Se describen los distintos esquemas de aprendizaje y el propósito de cada uno, posteriormente se profundiza en el papel que juegan los datos respecto al ML, así como el procedimiento que se debe seguir para obtener datos representativos y con pocas redundancias (filtrado y preprocesamiento). También se habla de la importancia que tiene la separación correcta de los datos en distintos subconjuntos. Se explican los tipos de algoritmos que existen, detallando las diferencias entre algoritmos de clasificación y regresión.

En la siguiente sección de este capítulo, se explica a detalle el funcionamiento de los algoritmos de ML que serán utilizados en capítulos posteriores de este trabajo, seguido de esto se mencionan las principales métricas y técnicas de evaluación de un algoritmo, mismas que se ocupan en los siguientes capítulos, también se habla de la distinción entre parámetros e hiperparámetros. En la última parte, se describe la relación y aplicación del ML con la industria petrolera, mencionando algunos antecedentes que han empleado esta herramienta, posteriormente se profundiza en ejemplos de la literatura que emplean el ML para la predicción de propiedades PVT ya que este es el tema central en este trabajo.

3.1. Introducción

El ML es un área del campo de la inteligencia artificial (AI), que se enfoca en algoritmos que pueden ajustarse automáticamente para predecir con cierto grado de exactitud un evento tomando como base una serie de datos de entrada. Estos algoritmos son desarrollados para implementarse en computadoras. La complejidad del modelo y el tiempo de cómputo dependen en gran medida del problema a resolver y la cantidad de datos que el algoritmo deberá manejar.

Los algoritmos de ML son muy versátiles por lo que pueden ser implementados en cualquier disciplina, ejemplos de esto son los sistemas de sugerencias de contenido en aplicaciones de *streaming* donde el algoritmo es capaz de reconocer los gustos del cliente para sugerirle contenido afín, otro ejemplo es el sistema de filtrado de correos de spam donde el sistema detecta y separa los correos no

deseados de los relevantes, el ML también se aplica para automatizar procesos, detecciones de fraude financiero, servicio de atención al cliente, reconocimiento de voz, sistemas de seguridad, pronóstico del clima e inclusive protección de la fauna.

¿Qué es un algoritmo? Hablar de ML es hablar de algoritmos, estos se puede entender como un conjunto de pasos ordenados que permiten resolver un problema. En ML, los algoritmos pueden ajustarse para que el modelo arroje resultados más precisos. De acuerdo con Mueller y Massaron, 2021, existen cinco escuelas de pensamiento en cuanto a ML se refiere, cada una presenta sus ventajas para resolver un problema específico.

1. **Simbolistas**, usan la lógica para llegar a conclusiones razonables.
2. **Conexionistas**, aluden que un algoritmo de ML es similar a las conexiones neuronales del cerebro humano.
3. **Evolucionarios**, basan su estrategia en la supervivencia del más apto, es decir, del algoritmo que mejor se ajuste con el resultado deseado.
4. **Bayesianos**, toman un enfoque estadístico para resolver problemas.
5. **Analogistas**, buscan encontrar patrones ocultos en la información para predecir un resultado.

Cada algoritmo se apega en cierta medida, a una o varias de estas corrientes de pensamiento, por ejemplo, un algoritmo de redes neuronales artificiales basa su funcionamiento en la corriente de los conexionistas, la familia de algoritmos de Naive-Bayes siguen las ideas de los bayesianos, entre otros.

3.1.1. Conceptos básicos

Variables (*features*)

También denominadas como características, son los parámetros o las magnitudes tomadas por el algoritmo para realizar una predicción, usualmente los datos que recibe el algoritmo se ordenan en vectores, cada uno representa una entrada de datos, es decir que cada valor de ese vector corresponde a un dato diferente, a este se le llama característica o variable. Por ejemplo, un algoritmo que predice si una persona dará click a un anuncio con base en el género, el tiempo de navegación en el sitio y su edad, sería un problema con tres variables o características.

Etiquetas (*labels*)

Para un problema de clasificación (explicado más adelante) se puede ver a una etiqueta como uno de los posibles valores que puede tomar la respuesta del algoritmo, es decir, es la respuesta asociada a las variables (datos) de entrada. Siguiendo con el ejemplo anterior, existen dos etiquetas posibles ya que el problema tiene dos resultados posibles, "sí" y "no", estos resultados dependen de los valores que tomen los datos de entrada.

Sobreajuste (*overfitting*)

El sobreajuste es el fenómeno que se da cuando el algoritmo no generaliza adecuadamente la magnitud que se quiere predecir, por el contrario, el algoritmo en vez de generalizar, se adapta muy bien a los datos que se le dan. Es decir, el algoritmo memoriza las respuestas obtenidas con los datos de entrenamiento por lo que al intentar usar el modelo con datos diferentes, este tiene un desempeño pobre. El sobreajuste generalmente sucede cuando se trata de minimizar en exceso el error en los datos de entrenamiento. Esto se ve reflejado con un bajo error en los datos de entrenamiento pero un alto error en los datos de prueba (estos conceptos se explican a detalle en las siguientes secciones).

Subajuste

Caso contrario al anterior, este modelo es demasiado simple por lo que no captura la esencia de los datos correctamente, debido a esto, el modelo no se ajusta ni a los datos de entrenamiento ni a los datos de prueba. Este problema suele darse cuando se cuenta con muy pocos datos para entrenar al algoritmo o cuando la cantidad de características en los datos es muy reducida, también puede darse cuando no se permite que el algoritmo se ajuste lo suficiente a los datos, es decir, el error tiene poca penalización.

Varianza

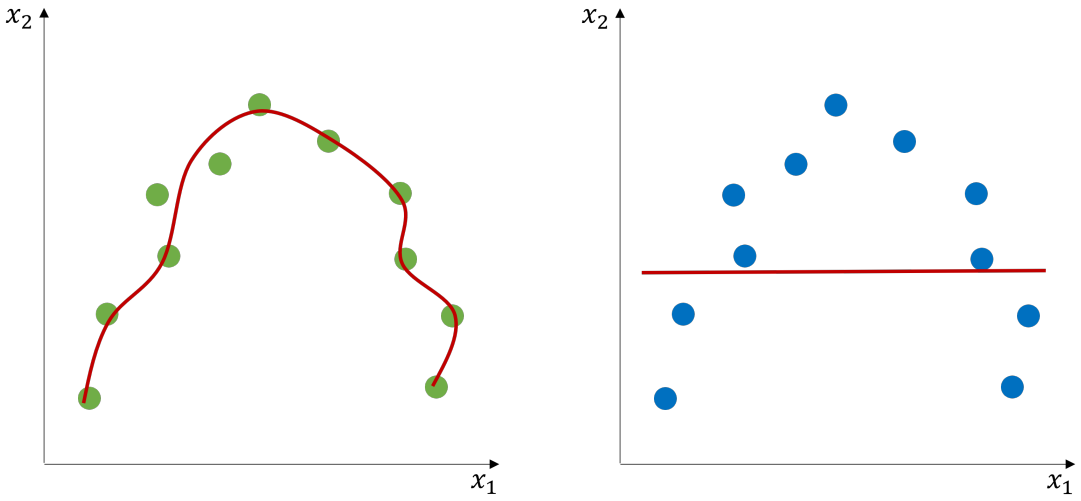
Es la desviación o variación de una predicción respecto a su valor real, es decir, cuando el algoritmo predice correctamente los datos de entrenamiento pero tiene un bajo desempeño en los datos de prueba. La ecuación 3.1 muestra como se obtiene este parámetro (Khan Academy, s.f.).

$$\sigma^2 = \sum (x_i - \mu_i)^2 \tag{3.1}$$

Sesgo (*bias*)

Un algoritmo está sesgado cuando tiene un mal desempeño tanto en los datos de entrenamiento como en los datos de validación, esto sucede cuando las suposiciones sobre el problema y los datos no son correctas y se ve reflejado con un error sistemático.

En las figuras 3.1 y 3.2a, se ejemplifican las definiciones de sobreajuste y subajuste. En primer lugar, la figura 3.1a, presenta un algoritmo que sobreajustó los datos y en vez de capturar la esencia de los datos, se sobreajustó a ellos, si el algoritmo se probara con otros datos, arrojaría resultados muy pobres ya que solo funciona bien con los datos de la figura. El caso contrario se tiene en la figura 3.1b donde el algoritmo no logró capturar el comportamiento de los datos y no tiene un buen desempeño al aplicarse. En consecuencia no tendrá un buen desempeño si fuese aplicado a datos distintos. Por último se presenta la figura 3.2a donde se presenta un algoritmo balanceado, donde se generaliza la estructura de los datos adecuadamente. Entonces, este algoritmo será capaz de realizar buenas predicciones en un set de datos distinto.



(a) Sobreajuste

(b) Subajuste

Figura 3.1: Ejemplo de sobreajuste (3.1a) y subajuste (3.1b)

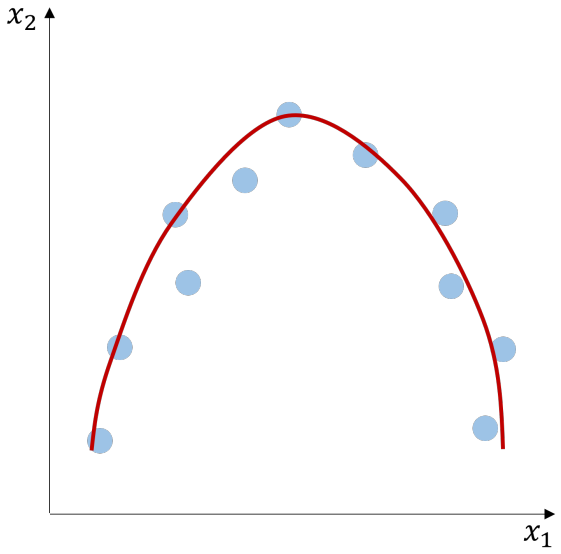


Figura 3.2: Ejemplo de un modelo correctamente ajustado, sin sobreajuste ni subajuste

3.2. Tipos de aprendizaje

La tarea de un algoritmo de ML consiste en pronosticar mediante un conjunto de datos de entrada o clases una respuesta o etiqueta esperada. De acuerdo con Shobha y Rangaswamy, 2018, los algoritmos de ML se dividen en cuatro categorías, éstas son esquematizadas en la figura 3.3 y mencionadas a continuación:

1. Aprendizaje supervisado
2. Aprendizaje no supervisado
3. Aprendizaje semi-supervisado
4. Aprendizaje con refuerzo

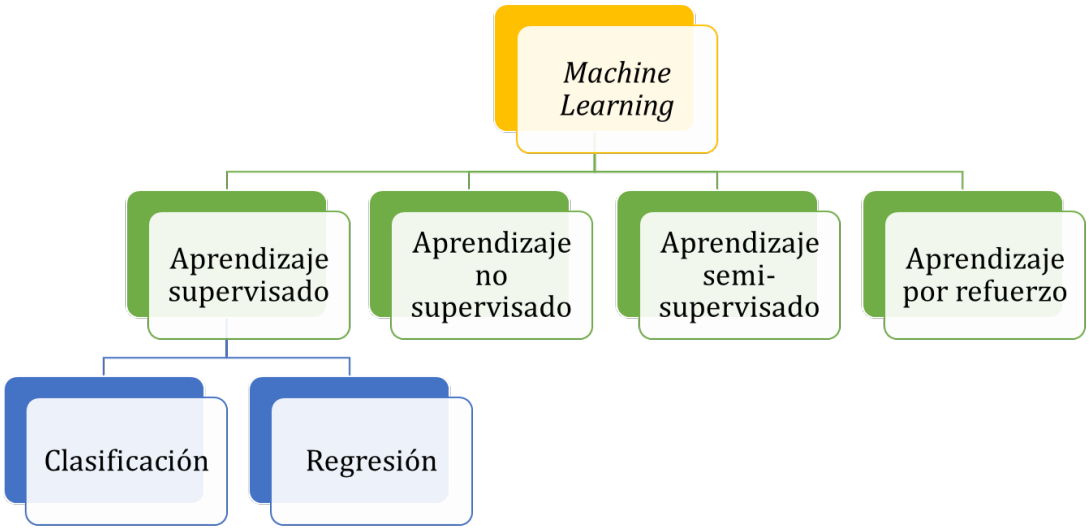


Figura 3.3: Tipos de aprendizaje, modificado de Shobha y Rangaswamy, 2018, p.199

3.2.1. Aprendizaje supervisado

Es aquel en el que el algoritmo tiene a su disposición tanto las señales de entrada como las respuestas, es decir, al algoritmo se le proveen datos de entrada y las etiquetas resultantes para un conjunto de datos, de esta manera el algoritmo trata de encontrar patrones en los datos de entrada con lo que generará un modelo que describa la relación entre los datos de entrada y sus etiquetas.

3.2.2. Aprendizaje no supervisado

En este tipo de aprendizaje se alimenta al algoritmo solamente con datos de entrada, omitiendo la respuesta, el algoritmo debe determinar los patrones en el conjunto de datos y la respuesta resultante por su propia cuenta. Este tipo de aprendizaje es útil para reestructurar datos y encontrar nuevas clases que sean útiles para algoritmos supervisados.

3.2.3. Aprendizaje semi-supervisado

De acuerdo con Shobha y Rangaswamy, 2018, este aprendizaje consiste en darle al algoritmo un conjunto de datos etiquetados y un conjunto de datos no etiquetados para que el algoritmo pueda etiquetar algunos de estos datos, el conjunto de datos no etiquetados es de un tamaño mucho mayor que el conjunto de datos etiquetados para que el algoritmo funcione efectivamente, de ser de tamaños similares ambos conjuntos, se opta por algoritmos que empleen otro tipo de aprendizaje.

3.2.4. Aprendizaje por refuerzo

Se refiere al aprendizaje donde al algoritmo se le dan datos de entrada sin la respuesta, en función de la respuesta obtenida con el modelo generado, el algoritmo es retroalimentado positiva o negativamente. Y el modelo se recalibra para obtener una mejor retroalimentación, este proceso se repite hasta que se genera un modelo lo suficientemente preciso; esto se asemeja al proceso de prueba y error.

3.3. Datos

Para que un algoritmo de ML sea efectivo, requiere de datos, la manipulación correcta de estos datos es parte vital de esta disciplina. Los datos son relativos a la variable que se desea predecir, estos tienen la función de entrenar al algoritmo para que este genere un modelo que pueda predecir un evento al recibir datos de entrada. Entre más datos sean dados al algoritmo en su fase de entrenamiento, este será más preciso al momento de hacer una predicción (Mueller y Massaron, 2021). La fase de entrenamiento consiste en darle ejemplos al algoritmo en forma de datos, con esto se genera una función que permite arrojar resultados similares a aquellos observados en los ejemplos. El conjunto de datos puede separarse en tres subconjuntos, datos de entrenamiento, de prueba y de validación, a continuación se menciona su propósito.

3.3.1. Datos de entrenamiento

El primer subconjunto es el más grande de los tres, este conjunto será usado por el algoritmo para ser entrenado, es importante que los datos que se le den al algoritmo sean datos reales, usualmente este subconjunto representa el 70 % del total del conjunto de datos. El algoritmo calibra sus parámetros internos (véase sección 3.5) con base a este subconjunto.

3.3.2. Datos de prueba

Este subconjunto normalmente es el 20 % del total de los datos y es usado una vez que se entrenó al algoritmo con los datos de entrenamiento. El algoritmo trata de predecir la salida esperada con

los datos de prueba y se compara con la salida real registrada, con lo que se puede observar que tan acertado es el modelo. Este subconjunto permite que el usuario calibre los hiperparámetros (véase sección 3.5) del algoritmo.

3.3.3. Datos de validación

El 10 % restante del total de los datos usualmente se usa como conjunto de validación, este set de datos se aplica al modelo ya entrenado y con hiperparámetros ajustados. Al usar un set de datos de prueba como herramienta para calibrar el modelo, se corre el riesgo de los parámetros de este modelo estén sesgados hacia este set de prueba, por esto se introduce este tercer subconjunto, si el modelo ha logrado capturar la estructura general de la propiedad a predecir, los resultados sobre el set de prueba y de validación deberán de ser comparables. La figura 3.4 muestra de manera gráfica la proporción típica de cada set de datos.

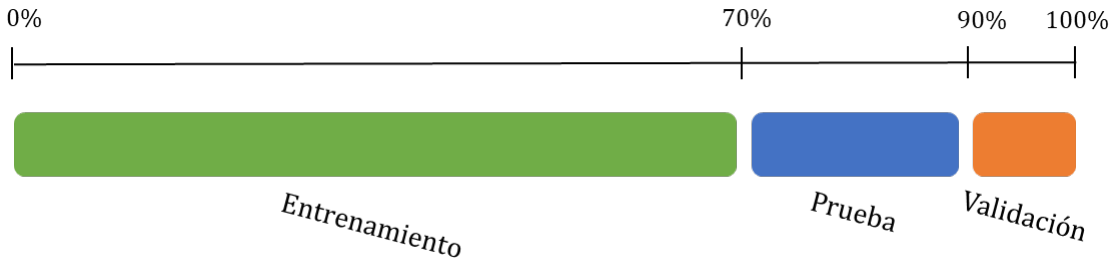


Figura 3.4: Proporción usual de cada subconjunto de datos

3.3.4. Filtrado de datos

Todo modelo generado mediante un algoritmo de ML requiere datos para "aprender", estos datos deben de representar adecuadamente el fenómeno que se quiere modelar, los datos crudos muchas veces no son representativos del fenómeno, esto es causado por variables ajenas al fenómeno que modifican la respuesta registrada en los datos. Dichas variables pueden ser ruido en los datos, errores sistemáticos, errores aleatorios al medir, datos incompletos, entre otros. Por ejemplo, los datos crudos obtenidos de una prueba de presión pueden tener interferencias debido al ruido, al funcionamiento deficiente en el sensor de presión o un error en su calibración. Para evitar esto, se debe preparar adecuadamente a los datos que se usarán para entrenar al algoritmo, en esta etapa se filtran y preprocesan los datos crudos para obtener un conjunto de datos que sea lo más representativo posible del fenómeno que se desea pronosticar.

A menudo los datos se encuentran incompletos, es decir, con datos faltantes, lo que hace prácticamente imposible que el algoritmo genere un modelo correcto. Para evitar esto se pueden descartar las series de datos que tengan faltantes. De acuerdo con Mueller y Massaron, 2021, dependiendo del algoritmo y la naturaleza de los datos puede optarse por:

1. Reemplazar el valor con uno fuera del rango normal (en algoritmos tipo árbol de decisión).
2. Reemplazar el valor con cero (en algoritmos de regresión), o características estandarizadas.
3. Reemplazar el valor por la media.
4. Interpolar el valor cuando es una función del tiempo.

3.3.5. Preprocesamiento de datos

Para optimizar el funcionamiento del algoritmo de ML a menudo se usan transformaciones en los datos que ayuden al algoritmo a converger a una solución más rápidamente o que ayuden a minimizar los errores en las predicciones.

Estas transformaciones usualmente reescalan los datos de cierta manera que también es de utilidad para sets de datos con *outliers* o valores atípicos (Buitinck y col., 2013). Las transformaciones de reescalamiento que serán de utilidad para este trabajo son:

1. Estandarización, consiste en restar la media a todos los valores y dividir esto entre la desviación estándar generando una distribución con la mayor parte de los valores entre -3 y 3, y se obtiene una distribución con media igual a cero y varianza unitaria.

   Existen dos variantes a la estandarización usual, donde los datos se reescalan dentro de cierto rango:

   $$x_{escalada} = \frac{x - \mu_x}{d.e.} \tag{3.2}$$

   a) Escalamiento mínimo-máximo, en este caso el rango de valores se encontrará dentro de un rango dado, usualmente entre cero y uno.

   $$x_{escalada} = \frac{x - x_{min}}{x_{max} - x_{min}} * \left[(x_{max} - x_{min}) + x_{min}\right] \tag{3.3}$$

   b) Escalamiento máximo absoluto, en este caso solamente se divide cada valor entre el valor máximo, por lo que el valor más alto una vez transformados los datos será de 1.

   $$x_{escalada} = \frac{x}{x_{max}} \tag{3.4}$$

2. Normalización, consiste en restar el valor mínimo de los datos y dividir esto entre la diferencia del valor más alto y el más pequeño, es decir, el rango de los datos.

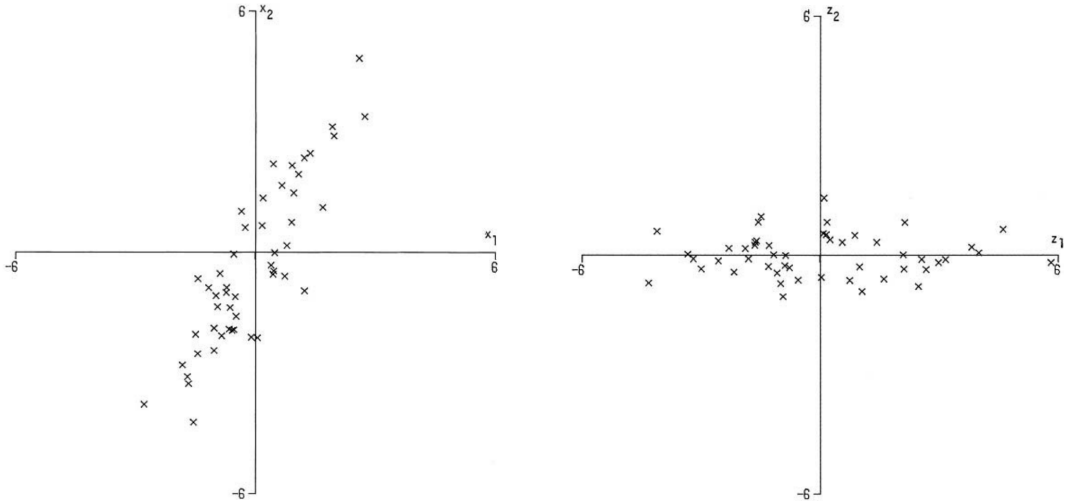
   $$x_{escalada} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{3.5}$$

Para que un algoritmo de ML obtenga buenos resultados, las características con las que se va a entrenar no deben correlacionarse entre sí. En la práctica, las características suelen estar correlacionadas, por lo que el algoritmo va a ser alimentado con características que tienen información redundante lo que disminuye el poder predictivo del modelo. Para evitar esto, se debe realizar una selección de características que den la mayor cantidad de información y estén lo menos correlacionadas entre sí.

3.3.6. Análisis de componentes principales

Una técnica para la selección de características es el "Análisis de componentes principales" o PCA por sus siglas en inglés, describir detalladamente esta técnica va más allá de los alcances de este trabajo por lo que se explicará brevemente su utilidad para la selección de variables.

Esta técnica proviene del análisis estadístico y permite reestructurar un conjunto de datos manteniendo su forma para eliminar la correlación entre sus componentes. Esta transformación ortogonal permite reducir la covarianza entre diferentes características y maximizar la varianza, además de esto, los datos son acomodados en orden, donde al inicio se colocan las características (o componentes) con la mayor varianza y al final aquellas con la menor varianza (Ramirez y col., 2017).



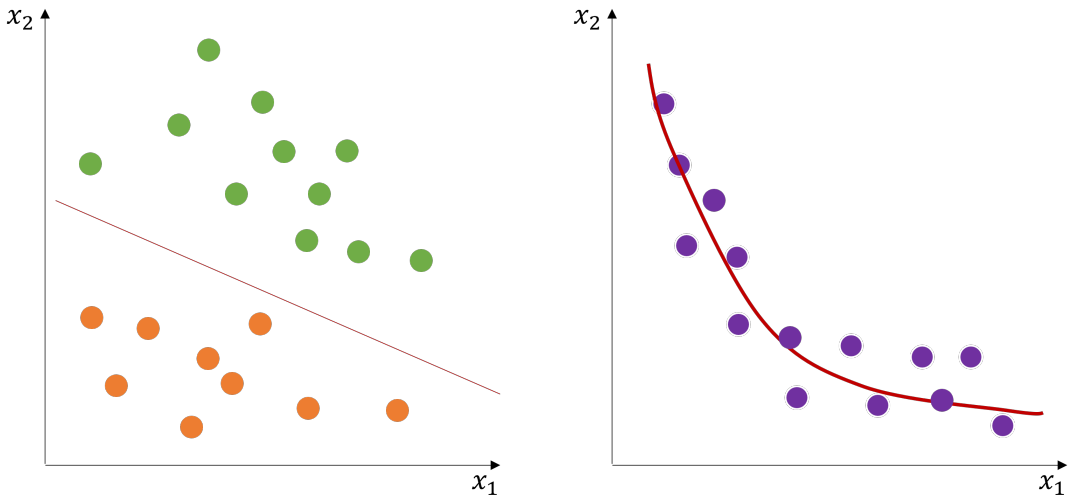
(a) (b)

Figura 3.5: Efectos de PCA para un set de 50 datos, en 3.5a se tienen las observaciones con dos variables x_1 y x_2, mientras que en 3.5b se tienen las mismas observaciones graficadas con respecto a sus componentes principales z_1 y z_2, tomado de Jolliffe, 2002, p.2

Teniendo en cuenta lo mencionado previamente, los primeros componentes son los que aportan más información para describir el fenómeno a predecir, esto permite eliminar los componentes con la menor varianza, porque aportan la menor cantidad de información. Así se reduce la dimensión del conjunto de datos, lo que ayudará al algoritmo a generar un modelo más preciso, y se reducirá el tiempo de cómputo, en la figura 3.5 se observan los efectos de esta transformación para un set de cincuenta datos.

3.4. Tipos de algoritmo

En este trabajo, se abordará un esquema de aprendizaje supervisado, es decir, aquel donde al algoritmo se le proporcionan datos y los resultados para generar un modelo, los algoritmos pueden dividirse en dos: clasificación y regresión (véase figura 3.6). A continuación se profundiza en cada una.



(a) Clasificación (b) Regresión

Figura 3.6: Clasificación y regresión, modificado de Liu, 2020, p.12

3.4.1. Clasificación

En este caso, la respuesta que el modelo predice es discreta y cae en un conjunto de categorías (etiquetas) o clases. Por ejemplo, predecir si un correo va a ser asignado a la carpeta de spam o no. La clasificación a su vez puede dividirse en tres subcategorías, binaria, multi-clase y multi-etiqueta (*multi-label*) (Liu, 2020, p.56).

Binaria

Aquí, la clasificación tiene solamente dos valores posibles de resultado como se muestra en la figura 3.6a. Un ejemplo de esto, es el caso donde se quiere predecir si a una persona le gustará una película, esta clasificación es binaria ya que solamente se puede tener como respuesta sí o no.

Multi-clase

En este caso, la clasificación tiene más de dos valores o clases diferentes (véase figura 3.7), un ejemplo es el reconocimiento de dígitos escritos a mano, donde los posibles valores van del cero al nueve.

Multi-etiqueta

A diferencia de los dos tipos anteriores donde cada clase es mutuamente excluyente, esta clasificación permite la asignación en varias categorías diferentes. Como ejemplo se encuentra la clasificación de películas en diferentes géneros, una película puede caer en la categoría de acción y de ciencia ficción al mismo tiempo. Usualmente se compone de pequeños clasificadores binarios (véase figura 3.8).

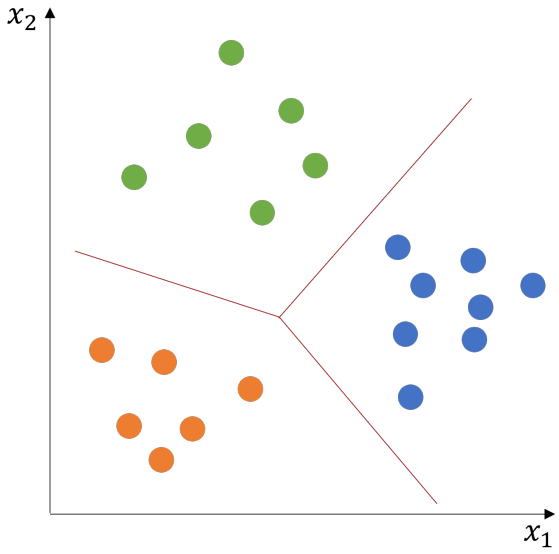


Figura 3.7: Clasificación multi-clase, modificado de Liu, 2020, p.59

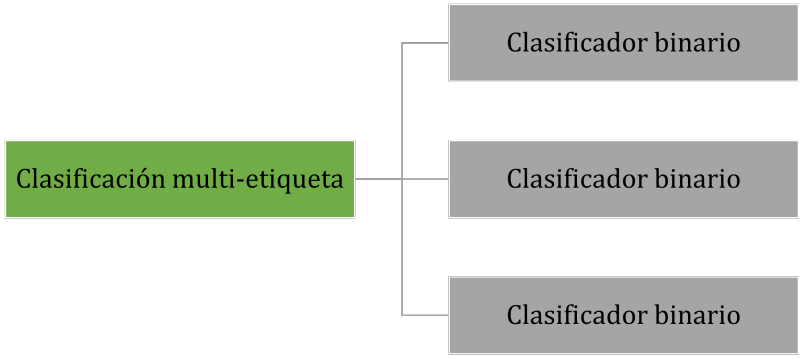


Figura 3.8: Clasificación multi-etiqueta, modificado de Liu, 2020, p.60

3.4.2. Regresión

En este caso, el modelo predice respuestas de valores continuos (numéricos). Esto se ejemplifica en la figura 3.6b. Un ejemplo es la predicción del precio del barril de aceite. Liu, 2020, divide el proceso de regresión en dos etapas, estas se mencionan a continuación y son esquematizadas en la figura 3.9.

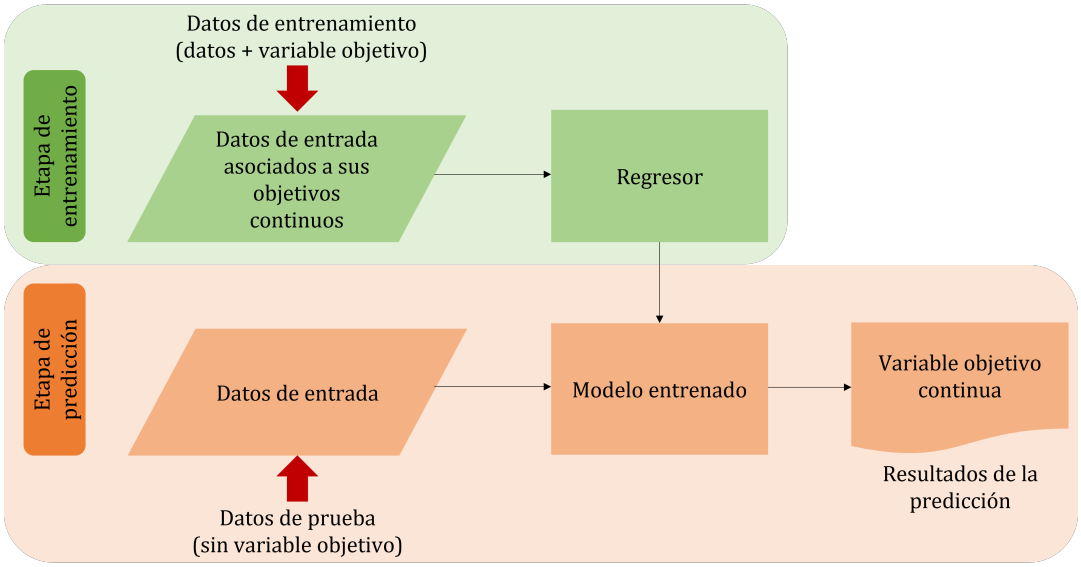


Figura 3.9: Etapas en un modelo de regresión, modificado de Liu, 2020, p.249

1. Primera etapa (de entrenamiento): Explorar las relaciones entre las observaciones o características y los objetivos, esto se denomina como la fase de entrenamiento.
2. Segunda etapa (de predicción): Usar los patrones encontrados en la primera fase para generar el objetivo de una futura observación, denominada como la fase de predicción.

La clasificación predice un valor discreto o una clase mientras que la regresión predice una cantidad continua; sin embargo, puede darse el caso donde se use un modelo de regresión para predecir un valor discreto y viceversa, tener un modelo de clasificación para predecir una cantidad continua. Un modelo de regresión estima un valor discreto, este valor se asigna si la predicción continua del modelo cae en cierto rango de valores. Por otro lado, un modelo de clasificación puede predecir un valor continuo en forma de probabilidad de que la predicción caiga en cierta categoría (Brownlee, 2017b). Ejemplos de algoritmos que pueden implementarse en problemas de clasificación como de regresión con pequeñas modificaciones en su estructura son, los árboles de decisión o las redes neuronales artificiales.

3.5. Hiperparámetros

Cada algoritmo funciona de manera diferente, porque se desarrolla con fundamentos matemáticos y parámetros propios de cada algoritmo. Para ajustar el modelo y optimizarlo es necesario realizar una calibración de sus hiperparámetros. Los cuales se definen como aquellos parámetros que el algoritmo no ajusta por si solo (Nyuytiymbiy, 2020). La diferencia entre parámetros e hiperparámetros es la siguiente:

Parámetros

1. Internos al proceso de aprendizaje del modelo.
2. Son estimados y ajustados de acuerdo a los datos que se ingresan al modelo.
3. Usualmente son guardados como parte del modelo ya entrenado.
4. Generalmente no son fijados por el usuario.
5. Forman parte de la formulación matemática del modelo.

Hiperparámetros

1. Externos al proceso de aprendizaje del modelo.
2. Usualmente ayudan a estimar los parámetros del modelo.
3. Dados por el usuario.
4. Su valor usualmente no puede ser estimado mediante los datos.

El ajuste de los hiperparámetros es un tema medular de cualquier problema de ML, cada problema es distinto, así como los datos que se tengan a disposición, es por esto que el ajuste de estos hiperparámetros es particular a cada problema. En algunos casos, suelen usarse reglas de dedo para aproximar el mejor valor de cierto hiperparámetro (Brownlee, 2017a).

En cada algoritmo, es usual tener más de dos o tres hiperparámetros, algunos con valores discretos y otros con un rango de valores continuos de los cuales se elige la combinación con la cual el modelo arroje los mejores resultados (Brownlee, 2017a). Debido a que la cantidad de combinaciones y complejidad del modelo aumenta de manera exponencial con cada hiperparámetro, existen herramientas que agilizan el proceso de búsqueda para la mejor combinación de hiperparámetros .

3.5.1. Búsqueda de malla

Existe una herramienta denominada como *Grid search* o búsqueda de malla y permite evaluar las distintas combinaciones de hiperparámetros que se deseen probar para un algoritmo dado. La mejor combinación será aquella que arroje la mejor puntuación de acuerdo con la métrica especificada (por ejemplo R^2 puede usarse como métrica de evaluación para problemas de regresión de acuerdo con Mueller y Massaron, 2021, p.170).

Esta herramienta se encuentra preestablecida en diferentes bibliotecas relacionadas con ML como las que se emplean en este trabajo, para el desarrollo de esta tesis se emplea simultáneamente una búsqueda de malla y una validación cruzada (véase figura 3.16), con el objetivo de evaluar cada combinación de los diferentes hiperparámetros de cada algoritmo.

Este tipo de herramientas es de gran utilidad ya que automatizan el proceso de búsqueda y calibración de hiperparámetros; sin embargo, el usuario debe de suministrar los diferentes valores que se evaluarán para cada hiperparámetro por lo que sigue siendo un proceso de prueba y error, donde se observa cuidadosamente el desempeño del algoritmo a medida que se modifica cada parámetro. Además, se debe analizar el peso que tiene cada parámetro tanto en el tiempo de cómputo (que crece exponencialmente), como en la calidad del modelo generado.

3.6. Algoritmos

Existe una gran cantidad de algoritmos de ML y, dependiendo del problema, los datos y recursos disponibles pueden ser más o menos eficientes. Usualmente cada familia de algoritmos tiene variantes que optimizan algún parámetro o simplemente extienden el alcance del algoritmo. En esta sección se mencionan los algoritmos que serán empleados en capítulos siguientes y, a juicio del autor, son algoritmos que tienen una filosofía de funcionamiento fácil de entender.

Una descripción detallada de los fundamentos matemáticos de cada algoritmo va más allá de los alcances de este trabajo; sin embargo, existe una gran cantidad de literatura disponible sobre el tema, se recomienda revisar los trabajos de Liu, 2020, Shobha y Rangaswamy, 2018 y Mueller y Massaron, 2021.

3.6.1. Redes neuronales artificiales

Este tipo de algoritmo está basado en un principio de funcionamiento muy simple, el de un perceptrón, concebido en 1957 por Frank Rosenblatt en el laboratorio de aeronáutica de Cornell, el

perceptrón es un algoritmo que busca encontrar el conjunto de valores de un vector w o vector de coeficientes mediante aproximaciones iterativas. Al multiplicar el vector de coeficientes con la matriz de características o datos de entrada y sumarle un término constante llamado "bias" se obtiene un vector o una matriz con los valores resultados esperados (Mueller y Massaron, 2021, p.176).

$$w = w + \eta * x(y - \hat{y}) \tag{3.6}$$

Esta es la ecuación que describe como los valores del vector de coeficientes se actualizan con cada iteración, (w) es el peso, (η) es el ritmo de aprendizaje, un valor constante que regula que tan grande es la magnitud de actualización del peso (w), tiene un rango de cero a uno, entre mayor es su valor, la actualización del vector se ve fuertemente impactada. Mientras que valores cercanos a cero limitan la capacidad de actualización del vector, (x) es el valor de entrada, mientras que $(y - \hat{y})$ es la diferencia entre el valor real de salida y el valor estimado (Mueller y Massaron, 2021, p.179).

El núcleo de una red neuronal es la neurona, al agrupar un conjunto de neuronas interconectadas se obtiene una red neuronal. Cada neurona recibe una señal de entrada, aplica una transformación y emite una señal de salida que es recibida por la siguiente neurona o la siguiente capa de neuronas sucesivamente hasta llegar a la última capa de neuronas.

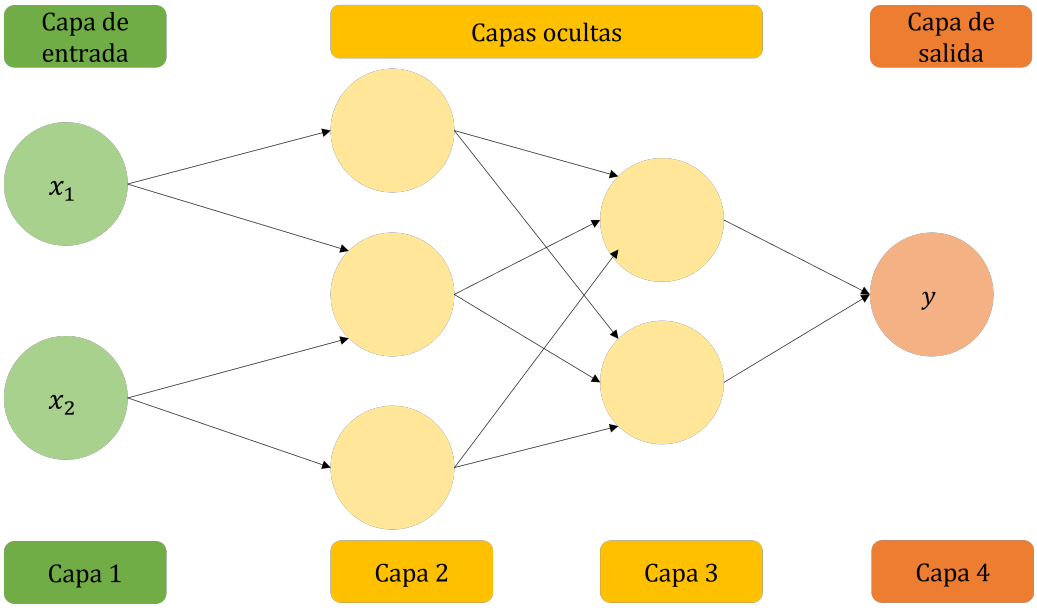


Figura 3.10: Capas en una red neuronal, en este ejemplo se tienen dos capas ocultas

El concepto de perceptrón es el principio con el que funciona una red neuronal, en este caso las neuronas que conforman a la red son una versión mejorada del perceptrón, mantienen el mismo principio de funcionamiento que éste pero añaden una función de activación, esto significa que una neurona no siempre va a enviar una señal de respuesta. La decisión de enviar o no una señal de

respuesta depende de la función de activación elegida. Esta particularidad le da una gran ventaja a estos algoritmos, porque las redes neuronales transforman los datos de una forma no lineal extendiendo su aplicación a problemas no lineales (Mueller y Massaron, 2021, p.272).

Una red neuronal está compuesta de neuronas agrupadas en capas, la primera capa de neuronas es la que recibe los datos de entrada del problema, esta primera capa es la capa de entrada o *input layer* donde se reciben los datos de entrada, las capas intermedias son denominadas como capas ocultas o *hidden layers* y la última capa es conocida como capa de salida o *output layer*, en esta se obtienen los resultados o predicciones. Esto puede observarse en el esquema de la figura 3.10.

Cada capa es un conjunto de neuronas que se encuentran interconectadas con las neuronas de la siguiente capa. Para algoritmos de clasificación binaria, en la última capa se tiene una única neurona donde el valor obtenido será la probabilidad de pertenecer a una de las dos clases. Para clasificación multi-clase se tendrán n neuronas donde n es el número de clases totales y cada neurona representará la probabilidad estimada de pertenecer a esa clase. Finalmente, para problemas de regresión se tiene una neurona en la última capa para cada variable a predecir.

Función de activación

La función de activación en una neurona es lo que le da capacidad a una red neuronal para tratar con problemas no lineales, es un filtro que se aplica a la señal antes de ser emitida por la neurona, es decir, una transformación de la señal antes de ser enviada a la siguiente capa (Mueller y Massaron, 2021, p.273). Existen diferentes funciones de activación, la elección de éstas al diseñar una red neuronal depende del problema a modelar. En la figura 3.11 se muestra el comportamiento de diferentes funciones de activación, estas funciones se detallan más a fondo a continuación.

Función identidad, la función más simple, en este caso es la función de identidad típica, con un intervalo de (-∞, +∞), puede observarse en la figura 3.11a y está descrita por la ecuación 3.7.

$$f(z) = z \tag{3.7}$$

Función logística, también llamada sigmoide, va en un intervalo de (0,1) está dada por la ecuación 3.8 y puede apreciarse su comportamiento en la figura 3.11b.

$$f(z) = \frac{1}{1 + e^{-z}} \tag{3.8}$$

Función Tanh, está dada en el intervalo (-1,1) por la ecuación 3.9 y se muestra en la figura 3.11c.

$$f(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} = \frac{2}{1 + e^{-2z}} - 1 \tag{3.9}$$

Función ReLU, o "rectified linear unit", su comportamiento se muestra en la figura 3.11d, dada por la ecuación 3.10.

$$f(z) = max(0, z) \tag{3.10}$$

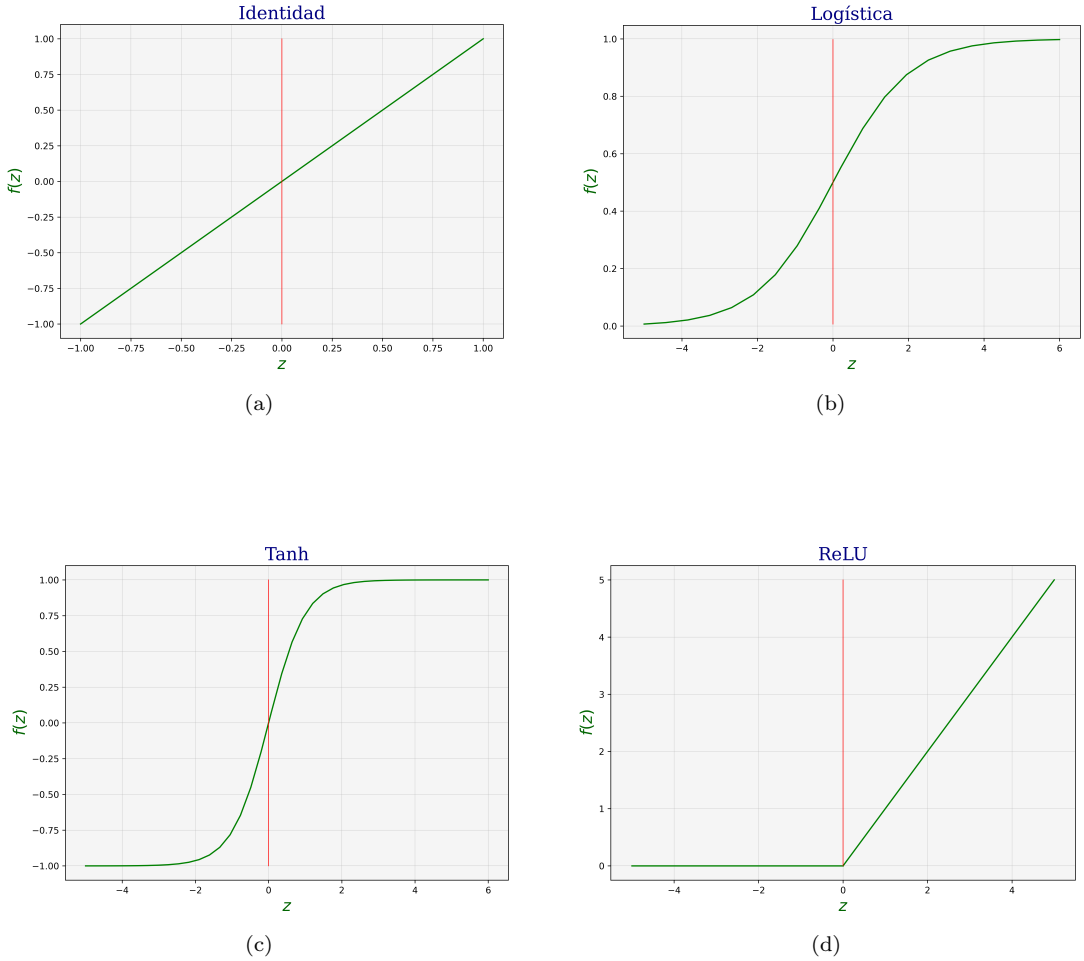


Figura 3.11: Diferentes funciones de activación, 3.11a modificado de Mate Labs, 2017 y 3.11b, 3.11c y 3.11d modificado de Liu, 2020, p.296

Retropropagación

También conocida como "backpropagation" es la manera en la cual una red neuronal ajusta los pesos en el modelo, el objetivo es reducir el error cuadrado medio (MSE), esto se realiza al calcular el gradiente Δw (véase ecuación 3.6) desde el final hacia el inicio de la red neuronal, es decir, el gradiente de la última capa se determina primero y el gradiente de la primera capa se determina al último. Los cálculos del gradiente en una capa son reutilizados para el cálculo del gradiente en la siguiente capa, de ahí el término de retropropagación (Liu, 2020, p.298).

3.6.2. Árboles de decisión

Este algoritmo tiene la ventaja de ser bastante intuitivo y versátil para aplicarse a problemas de clasificación de todo tipo. Su uso ha sido extendido para aplicarse a problemas de regresión (Mueller y Massaron, 2021). Al observar los datos de entrenamiento, el algoritmo genera reglas para aproximarse a los resultados de estos, ya sean etiquetas o valores numéricos continuos para problemas de regresión.

De acuerdo con Shobha y Rangaswamy, 2018, un árbol de decisión predice un resultado al aprender reglas de decisión, toma los datos de entrada y los separa mediante una regla de decisión. Posteriormente cada set de datos separados, vuelve a ser separado mediante un nuevo criterio de decisión. El árbol comienza con el nodo raíz, después se ramifica varias veces hasta llegar al nodo final, también llamado hoja, la figura 3.12 muestra un ejemplo de la estructura de un árbol de decisión, mientras que en la tabla 3.1 se resumen las principales ventajas y desventajas de este tipo de algoritmos:

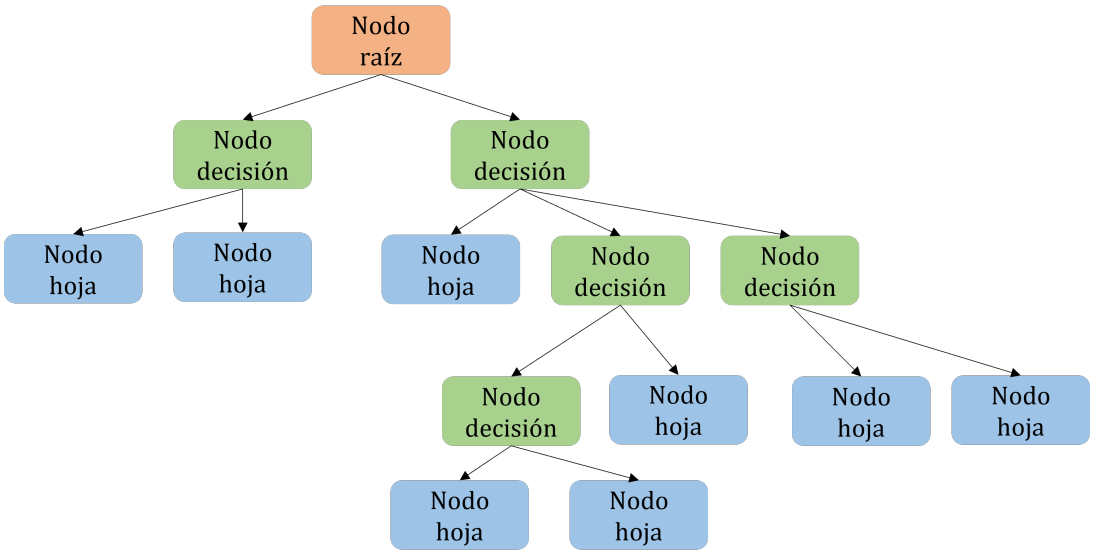


Figura 3.12: Ejemplo de la estructura de un árbol de decisión

Tabla 3.1: Ventajas y desventajas de un árbol de decisión

Ventajas	Desventajas
Fácil interpretación y representación	Sobre ajustan los datos fácilmente si no se limita el crecimiento de las ramificaciones
Bajo costo computacional	Afectado fuertemente por la varianza en los datos, pequeños cambios en los datos generan un árbol de decisión diferente
Manejo eficiente de datos y no requiere de normalización	Un solo árbol de decisión puede no generalizarse adecuadamente a la estructura de los datos
Capaz de manejar relaciones no lineales en los datos	Se pueden generar árboles sesgados si predomina una categoría en los datos
Son flexibles a diferentes tipos de problemas	No son óptimos para extrapolación

Random Forest

En este trabajo se usa una variación de los árboles de decisión, el algoritmo *Random Forest* (bosque aleatorio) debe su nombre al hecho de que implementa una gran cantidad de árboles de decisión, por otro lado, ese aleatorio porque los datos de entrenamiento que recibe cada árbol, en vez de alimentar a cada árbol con el set completo de datos de entrenamiento, los datos son separados aleatoriamente en subconjuntos de datos diferentes para cada árbol. Además de esto, las características en cada árbol son elegidas aleatoriamente para cada uno, finalmente, los resultados de cada árbol de decisión se combinan para generar el modelo final (Liu, 2020, p.166). Con esta separación aleatoria de los datos para alimentar a cada árbol se reduce el efecto negativo de la varianza en un árbol de decisión, además, se aprovecha el bajo costo computacional de generar un solo árbol de decisión.

Al aplicar este algoritmo a problemas de clasificación, se usa una técnica de votación para elegir el resultado predicho,y conocer qué tan bien ajustado está el modelo. Se puede observar el porcentaje de árboles que predijeron la categoría correcta; entre mayor sea el porcentaje, se tiene un mayor ajuste. En problemas de regresión, el algoritmo promedia los resultados de cada árbol, para determinar el ajuste del modelo y la precisión de las estimaciones, se puede observar la desviación estándar (Mueller y Massaron, 2021, p.322). En la figura 3.13 se esquematiza un modelo de *Random Forest*.

Extremely Randomized Trees

Esta variación de *Random Forest* también usa árboles de decisión como estimadores base, la diferencia clave en este tipo de algoritmos recae en el modo en que las ramificaciones son determinadas. Para este, se elige un subconjunto de propiedades para cada árbol de decisión, la elección de la división se da con base en el umbral de decisión que tenga la mayor discriminación. En *Extremely Randomized Trees*, esta división se realiza de manera aleatoria para cada propiedad y el mejor umbral de división es el que se toma como el criterio de división (Pedregosa y col., 2011). Por la

manera en la que funciona este algoritmo, reduce la varianza del modelo a expensas de aumentar ligeramente el sesgo.

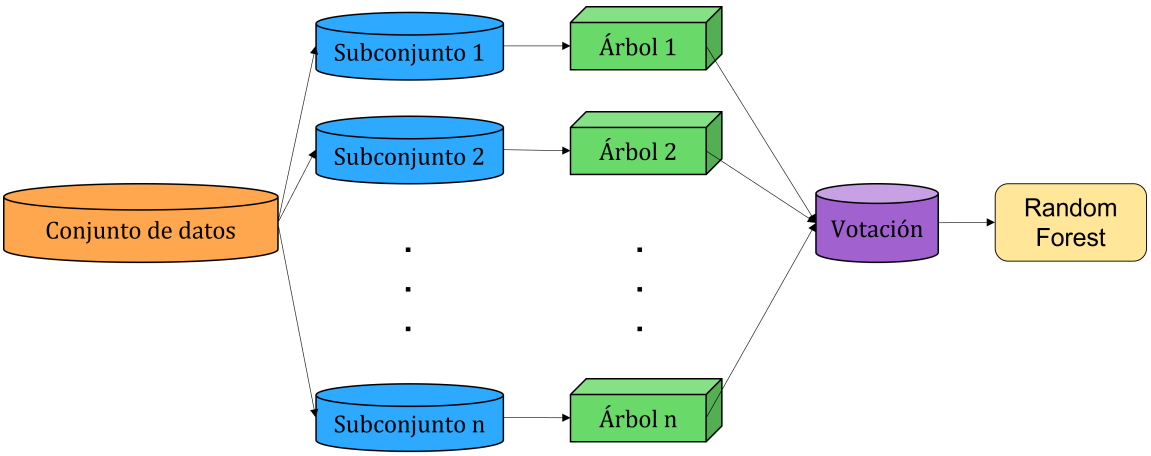


Figura 3.13: Estructura de un modelo *Random Forest*, modificado de Liu, 2020, p.170

3.6.3. Máquina de vectores de soporte

O *Support Vector Machine* (SVM), para este algoritmo enfocado a un problema de clasificación, se genera un conjunto de hiperplanos en un espacio dimensional mayor al de los datos. Dichos planos los datos de tal manera que los separan de la forma más óptima para las diferentes clases (Liu, 2020, p.120). Los hiperplanos son de $n-1$ dimensiones siendo n la cantidad de dimensiones de los datos de entrada.

La elección del hiperplano óptimo es aquel que maximiza la distancia entre sí mismo y los puntos cercanos a este. Estos puntos más cercanos al hiperplano se denominan **vectores de soporte** (Pedregosa y col., 2011).

En la figura 3.14 se muestra un ejemplo de este algoritmo para un problema de clasificación binaria. Para problemas de clasificación multi-clase o de regresión, la representación gráfica del algoritmo se vuelve muy compleja.

Este tipo de algoritmos se puede extender fácilmente a problemas de regresión, además de tener muchas variantes, por ejemplo $\nu - SVM$, es un algoritmo donde se calcula de manera alternativa el número de vectores de soporte y los errores en la clasificación de las observaciones (para problemas de clasificación) (Pedregosa y col., 2011).

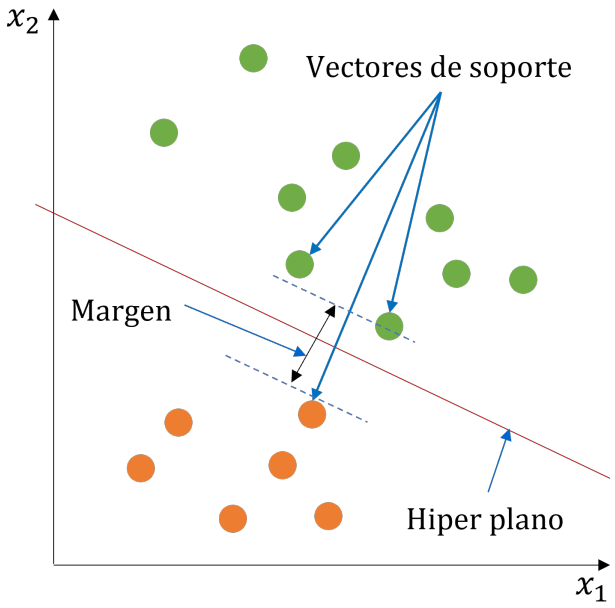


Figura 3.14: Ejemplo de vectores de soporte para un problema de clasificación binaria, modificado de Liu, 2020, p.94

3.6.4. *K-Nearest Neighbors*

Abreviado como *KNN*, este algoritmo tiene como objetivo encontrar cierto número de muestras de entrenamiento cercanas al punto que se quiere predecir y promediar estas observaciones para hacer una predicción, es decir, toma los puntos vecinos y los usa para obtener una predicción que sea similar a los resultados de los puntos vecinos (Pedregosa y col., 2011). El número de observaciones vecinas a tomar en cuenta es un parámetro clave para este tipo de algoritmos.

Existe una variación llamada *Radius Neighbors* donde el parámetro clave no es la cantidad de puntos vecinos, sino la distancia a la cual los puntos serán tomados en cuenta, es decir, el radio desde el punto que se desea predecir. La forma de medir la distancia también es especificada, como ejemplo puede ser distancia *euclidiana*, *Minkowski* o *Chebyshev* (Pedregosa y col., 2011).

Son útiles tanto para tareas de aprendizaje supervisado y no supervisado, para problemas de clasificación y regresión. Son rápidos de entrenar pero lentos al momento de realizar predicciones, debido a que muchos cálculos se realizan en esta etapa (Mueller y Massaron, 2021).

En la figura 3.15 se muestra un ejemplo de este algoritmo donde se tiene un problema de clasificación binaria. Se desea estimar si el punto gris es parte de la clasificación de puntos azules o amarillos. Puede observarse que al considerar los tres puntos más cercanos ($k = 3$), la estimación resultante indica que el punto desconocido forma parte de la clasificación azul, por otro lado, cuando se consideran los nueve puntos vecinos más cercanos, el punto gris se asigna a la categoría amarilla.

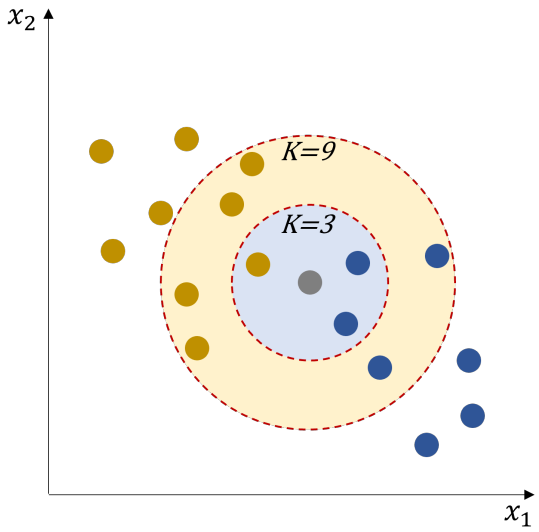


Figura 3.15: Ejemplo de *K-Nearest Neighbors* para clasificación binaria, con $k = 3$ el punto gris se clasifica como parte de los puntos azules, para $k = 9$, el mismo punto es clasificado como parte de los puntos amarillos, modificado de Chouinard, 2022

3.7. Evaluación del algoritmo

Para elegir el algoritmo óptimo para un problema, se debe evaluar su desempeño por medio de diferentes técnicas respecto al funcionamiento del algoritmo. Dependiendo del tipo de problema (clasificación o regresión), el método para evaluar el desempeño del modelo será diferente. A continuación se resumen las principales métricas para la evaluación de un algoritmo de ML de acuerdo con Shobha y Rangaswamy, 2018. Seguido de esto, se detallan las métricas usadas para la evaluación del algoritmo de ML que serán de utilidad para este trabajo.

1. Exactitud (*accuracy*)
2. Validación cruzada *k-fold*
3. Validación cruzada *k-fold* estratificada
4. Error medio absoluto
5. Error medio cuadrado
6. Error relativo
7. Coeficiente de correlación

3.7.1. Validación cruzada *k-fold*

Este método consiste en dividir el conjunto de datos de entrenamiento en k subconjuntos, el propósito de esto, de acuerdo con Shobha y Rangaswamy, 2018, es aumentar la cantidad de datos usados para validación sin reducir el tamaño del set de datos de prueba.

Al estar dividido en k subconjuntos, se emplean $k-1$ subconjuntos como datos de entrenamiento y el subconjunto restante es empleado para probar el algoritmo, este procedimiento se repite k veces, usando un subconjunto diferente como el set de datos de prueba. En la figura 3.16 se esquematiza un ejemplo de validación cruzada con $k = 5$.

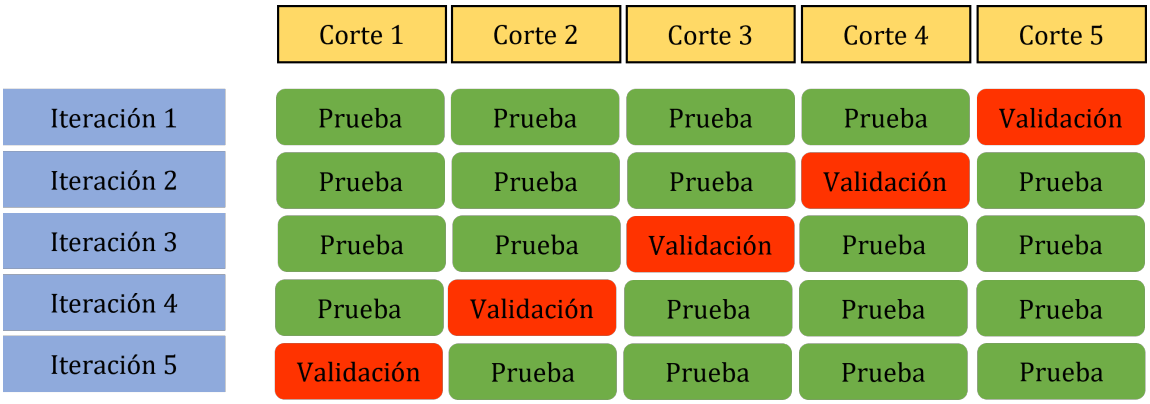


Figura 3.16: Validación cruzada con $k = 5$, modificada de Mueller y Massaron, 2021, p.166

Los datos se distribuyen de manera aleatoria en cada subconjunto, cada iteración calcula el error, por lo tanto, el error total es el promedio del error en cada iteración. De acuerdo con Mueller y Massaron, 2021, este método funciona adecuadamente independientemente del tamaño del conjunto de datos, por lo que se puede aplicar a sets de datos pequeños y grandes. Además, el impacto de que una iteración no tenga la misma distribución que las demás no es significativo ya que solo se usará una vez como conjunto de prueba. La principal ventaja de emplear este método es que se maximiza la utilidad del conjunto de datos ya que todos los datos se emplean para el aprendizaje del algoritmo.

Una desventaja del método es que la variación en el error con cada iteración puede ser muy distinta, por otro lado, no es posible aplicar este método en datos ordenados ya que los subconjuntos se generan con una distribución aleatoria. Un ejemplo de esto es cuando se trabaja con una serie de datos en función del tiempo (Shobha y Rangaswamy, 2018, p.206).

3.7.2. Error medio absoluto

MAE o *mean absolute error*, cuantifica el valor medio del error absoluto, en la misma escala que la variable objetivo. Es útil para determinar qué tan cercanas son las predicciones a los valores

reales (Liu, 2020, p.283). Está representada con la ecuación 3.11 (Fernández, s.f.).

$$MAE = \frac{\sum_i^n x_{real} - x_i}{n_{muestras}} \tag{3.11}$$

3.7.3. Error medio cuadrado

MSE o *mean squared error*, es el valor medio del error elevado al cuadrado. En ocasiones se calcula la raíz cuadrada de este valor para obtener un valor en la misma escala que la variable objetivo, a éste se le denomina "root mean squared error" o $RMSE$ (Liu, 2020, p.283). Se muestra en la ecuación 3.12 (Fernández, s.f.).

$$MSE = \frac{\sum_i^n (x_{real} - x_i)^2}{n_{muestras}} \tag{3.12}$$

3.7.4. Error relativo

Es el cociente de la diferencia entre el valor estimado y el valor real, dividido por el valor real, y puede verse su formulación matemática en la ecuación 3.13.

$$e_{rel} = \frac{x_{real} - x_i}{x_{real}} \tag{3.13}$$

3.7.5. Coeficiente de correlación, R^2

Es el coeficiente de correlación, indica que tan bien ajustado se encuentra el modelo respecto al comportamiento real de las variables, va de cero a uno, donde el valor más cercano a uno indica un mejor ajuste del modelo. Mientras que valores cercanos a cero indican que el modelo tiene un ajuste deficiente de los datos (Liu, 2020, p.283). Descrita por la ecuación 3.14 (Buitinck y col., 2013).

$$R^2 = 1 - \frac{\sum_i^n (x_i - x_{real})^2}{\sum_i^n (x_i - \mu_x)^2} \tag{3.14}$$

3.8. *Machine Learning* en la ingeniería petrolera

Como se mencionó anteriormente, el ML es un área relativamente nueva de conocimiento, por lo que su aplicación en otras ramas de la ingeniería ha sido gradual. Es necesario profundizar en trabajos de investigación que relacionen el ML con la industria petrolera y ciencias afines así como en la predicción de propiedades PVT, ya que es un tema medular en este trabajo. A continuación, se mencionan algunos ejemplos.

La aplicación de algoritmos de ML a problemas de reconocimiento de imágenes es bastante reconocida y con resultados notables, el trabajo de Guillen-Rondon y col., 2019, aprovecha esto al

implementar una variante del algoritmo de red neuronal llamado red neuronal convolucional (*Convolutional Neural Network*, CNN) para procesar imágenes sísmicas 2D con el propósito de detectar trampas estructurales (pliegues anticlinales), con acumulaciones de hidrocarburos. En el artículo de Odi y Nguyen, 2018, se plantea el uso de un modelo de *deep learning* (aprendizaje profundo), que es un algoritmo derivado de redes neuronales para la predicción de facies geológicas, con lo que se pretende reducir el tiempo necesario para esta etapa de prospección y reducir el factor humano en errores de interpretación. Otro ejemplo, es el que desarrollan Bandura y col., 2018 en su artículo donde aplican modelos de aprendizaje no supervisado para la agrupación de imágenes sísmicas. Además, investiga, la relación de algoritmos de aprendizaje supervisado para la clasificación de la litología con base en atributos sísmicos. En el área de perforación, Mal y col., 2022, analizan la aplicación de algoritmos de redes neuronales y árboles de decisión para reducir el riesgo de atrapamiento de tubería en operaciones de perforación, mediante redes neuronales y árboles de decisión. Los trabajos de Rogulina y col., 2022 y Jayeola y col., 2022 aplican modelos de ML al área de producción, enfocados a registros de producción y análisis de curvas de declinación, respectivamente. El ML también tiene ejemplos de aplicación en otras áreas de la industria, como la predicción de precios *spot* para el gas natural donde, de acuerdo a Ogwu y col., 2022, se aplican distintos algoritmos que abarcan desde redes neuronales artificiales, pasando por árboles de decisión y máquina de vectores de soporte, hasta algoritmos de *Gradient Boosting*. El aseguramiento de flujo es otra área de oportunidad donde se ha investigado acerca de aplicaciones del ML, el trabajo de Ugoyah y col., 2022, se enfoca en la aplicación de árboles de decisión para predecir la precipitación de sulfato de Bario ($BaSO_4$) y carbonato de calcio ($CaCO_3$); otro ejemplo en esta área es el artículo de Odutola y col., 2022, donde se emplean distintos algoritmos de regresión para estimar el riesgo de formación de hidratos de gas en tuberías para yacimientos no convencionales. En relación a yacimientos no convencionales, Palmer y Gu, 2022 proponen el uso del ML para estimar cuantitativamente de la relación entre cambios en propiedades de gas de lutitas con la propagación de fracturas hidráulicas. Todos estos son algunos ejemplos de la literatura donde el ML se ha aplicado a distintas áreas de la ingeniería petrolera con resultados que aumentan la confianza en la aplicación del ML a problemas actuales de la industria.

Para aplicaciones relacionadas con la estimación de propiedades PVT se resumen los siguientes trabajos disponibles en la literatura.

Yang y col., 2020, proponen el uso de 4 algoritmos diferentes para predecir la presión en el punto de burbuja. Los datos de entrada se dividieron en 3 conjuntos, el primero comprende la temperatura de yacimiento, relación gas-aceite, relación de solubilidad, densidad relativa del gas y densidad API, el segundo conjunto consiste de temperatura de yacimiento, porcentajes composicionales de $C_1 - C_7^+$ y de gases contaminantes N_2, H_2S y CO_2. Finalmente, el tercer conjunto es la unión de ambos conjuntos. Como se mencionó, se entrenaron cuatro algoritmos diferentes:

1. *XGBoost*
2. *LightGBM*
3. *Random Forest Regresor*
4. Perceptron multi capas (*MLP regressor*)

Adicionalmente, se empleó un algoritmo *Super Learner* también conocido como *Stacking Ensemble* donde básicamente se apila un conjunto de algoritmos diferentes (en este caso se emplean los cuatro algoritmos mencionados previamente) para predecir un resultado. La ventaja de este método es que permite reducir el error sustancialmente al integrar diferentes algoritmos en uno solo. Se empleó cada set de datos con cada uno de los algoritmos presentados, además de implementar una validación cruzada *k-fold*. Se observó que al usar el primer conjunto de datos, el algoritmo *Super Learner* tiene el mejor desempeño, para el segundo conjunto de datos. *MLP regresor* se desempeña ligeramente mejor que el algoritmo *Super Learner* y al tratarse del tercer conjunto de datos, *Super Learner* tiene el mejor desempeño.

Onwuchekwa, 2018, realiza una comparación entre diferentes algoritmos, *K-Nearest Neighbors* (KNN), *Support Vector Regression* (SVR), *Kernel Ridge Regression*, *Random Forest*, *Adaptive Boosting (Adaboost)* y *Collaborative filtering*, para la predicción de p_b, B_o y μ_o. El set de datos comprende información de 296 yacimientos de aceite y 72 de gas del delta del Níger, estos datos incluyen: profundidad, presión inicial, temperatura de yacimiento, relación gas-aceite, densidad API, densidad relativa del gas, viscosidad del aceite muerto, entre otras. El autor toma estos datos para estimar la presión de burbuja y el factor de volumen de formación del aceite así como la viscosidad del aceite; para este último caso, cada algoritmo arrojó resultados que superan a las correlaciones contra las que fueron comparados (M. Standing, 1947, Vasquez y Beggs, 1980 y Beggs y Robinson, 1975). En segundo lugar, se estimó la presión de rocío y el factor de volumen de formación del gas. En este caso el algoritmo de *Adaboost* mostró buenas estimaciones para ambos parámetros, mientras qué, los demás algoritmos mostraron un desempeño deficiente.

Ramirez y col., 2017, proponen un modelo de ML para estimar las propiedades PVT: p_b y B_o en función de la densidad API, relación de solubilidad, temperatura de yacimiento y densidad relativa del gas. Los datos empleados para este trabajo son una recopilación de datos publicados por diferentes autores de yacimientos en el Medio Oriente, Malasia y el Mar del Norte, con esto se generó un set con 400 datos. Para la estimación de las propiedades mencionadas primero se transforman los datos mediante la técnica PCA discutida previamente en este capítulo, posteriormente se desarrollaron dos modelos (uno para estimar p_b y otro para estimar B_o) mediante redes neuronales artificiales (*Artificial Neural Network*, ANN) con retroalimentación. Al comparar los resultados de ambos modelos con las correlaciones más populares para predecir estas propiedades (M. Standing, 1947, Vasquez y Beggs, 1980, Glasø, 1980, Al-Marhoun, 1988, Dokla y Osman, 1992, Petrosky y Farshad, 1993, Kartoatmodjo y Schmidt, 1994) se observó que el modelo de redes neuronales superaba en precisión a todas las correlaciones.

Numbere y col., 2013, emplean un set de 1246 datos del delta del Níger, de los cuales se toman 250 para validación de resultados; este set comprende temperatura, densidad relativa de ambas fases y relación de solubilidad con lo que se pretende predecir la presión de burbuja por medio de una red neuronal artificial. La arquitectura de la red consistió en una capa oculta con cuatro neuronas y la capa final solo se componía de una neurona representando el valor de la estimación de p_b. La métrica empleada en este trabajo es denominada *Rank*, esta métrica proviene del trabajo desarrollado por Ikiensikimama y Ogboja, 2009 en relación a la predicción de propiedades PVT. En este trabajo, la red neuronal es capaz de superar a las correlaciones empíricas tanto cuantitativa como cualitativamente.

Osman y col., 2001, presentan un modelo de ANN para predecir B_{ob} desarrollado a partir de un set de 803 datos de campos del Medio Oriente, Malasia, Colombia y el Golfo de México. De estos datos, 403 se emplearon para entrenamiento, 200 para validación cruzada y 200 como datos de prueba. El modelo incorpora una técnica de retropropagación (*backpropagation*) para reducir el error del algoritmo. El modelo incorpora cuatro neuronas en la primera capa (una para cada dato de entrada), cinco neuronas en la segunda capa y una neurona en la tercera capa que arroja el valor estimado de B_{ob}. Los resultados se compararon contra diferentes correlaciones (M. Standing, 1947, Vasquez y Beggs, 1980, Glasø, 1980, Al-Marhoun, 1988, Al-Marhoun, 1992), mediante un análisis estadístico donde se muestra que el modelo desarrollado tiene un mejor desempeño que todas las correlaciones empleadas, además de que se verificó que el modelo fuera físicamente correcto y estable sin sobre ajustar los datos.

Varotsis y col., 1999, desarrollan un modelo de ANN con retropropagación denominado *PVT expert®*, el modelo toma como datos de entrada la composición del aceite, la presión de burbuja, la temperatura de yacimiento, la densidad en la presión de burbuja, la viscosidad del aceite muerto, y las densidades de aceite y gas. Para condensados se toman como datos de entrada la composición del aceite, presión de rocío, temperatura de yacimiento, factor de desviación en la presión de rocío, relación gas-aceite y densidades de ambas fases a condiciones de superficie. Con el objetivo de lograr un mejor desempeño del modelo se transformaron los datos mediante un enfoque de "curva tipo" y un enfoque de "valor de referencia". El modelo usó un conjunto de 650 datos con los parámetros de entrada ya mencionados para predecir las siguientes propiedades, para el aceite: $B_o, RGA, \rho_o, \gamma_g, z$ y μ_o. Para los condensados: γ_g, z, fluido producido acumulado (% mol) y porcentaje de líquido retrógrado. Para la red neuronal se usaron dos capas ocultas, variando el número de neuronas en cada capa hasta obtener la estructura con los resultados más óptimos y con menor error.

Como se pudo apreciar en la revisión de la literatura previamente presentada, existen diversos autores que han utilizado el ML para predecir o resolver distintas propiedades y problemas relacionados a los hidrocarburos, con lo que se puede comprobar la expansión y utilidad de estas herramientas computacionales en la ingeniería petrolera, específicamente para la estimación de propiedades PVT.

Capítulo 4

Metodología

En este capítulo se explica a detalle el procedimiento seguido a lo largo de este trabajo. Primero se habla brevemente del software empleado para implementar los algoritmos encargados de predecir las variables objetivo B_{ob} y p_b, después, se describen los modelos de *Machine Learning* (ML) que serán implementados. Posteriormente, se describe el proceso de adquisición de los datos con los que se va a trabajar, se explican los pasos seguidos para el filtrado de datos y a continuación se habla del procedimiento para generar un set de datos más completo, obteniendo datos mediante el uso de correlaciones empíricas, así como datos obtenidos mediante la aplicación de un algoritmo de ML. Así mismo, se muestran de manera gráfica, los datos originales y los datos generados. Además, de una descripción gráfica del conjunto de datos de entrada definitivo empleado en el siguiente capítulo.

En la última parte del capítulo, se explica en qué consiste la metodología para predecir las dos variables objetivo, presión de burbuja (p_b) y factor de volumen de formación del aceite a la presión de burbuja (B_{ob}) y se mencionan los distintos casos en los que serán aplicados los algoritmos de ML. El desarrollo de esta metodología se ilustra con diagramas de flujo que permiten entender de una mejor manera las explicaciones y procedimientos desarrollados.

4.1. Software empleado

Para el desarrollo de este trabajo se eligió el lenguaje de programación de alto nivel denominado Python ya que éste posee varios atributos que lo destacan entre otros lenguajes para los propósitos de este trabajo.

Este lenguaje tiene una sintaxis bastante sencilla de entender ya que simula el lenguaje humano, por esta razón es apto para programadores principiantes, sin embargo, debido a su versatilidad, es idóneo para ejecutar tareas complejas, así que se mantiene como un lenguaje amigable tanto con programadores principiantes como avanzados. Además, es de código abierto por lo que cualquier persona lo puede usar y, posee una amplia variedad de módulos que expanden la capacidad del software a áreas más especializadas como el ML.

Existen bibliotecas o paquetes que permiten expandir las capacidades de Python para resolver problemas más específicos o especializados, en este trabajo se hace uso de varias bibliotecas, tanto para la aplicación de algoritmos de ML, como para la lectura, escritura, manipulación y visualización de datos. A continuación se describe la paquetería usada así como la versión de cada una.

***Pandas*,** esta biblioteca es empleada principalmente para la lectura, extracción y escritura de ficheros de archivos externos, principalmente aquellos archivos en formato (".xlsx" y ".csv"). Ofrece herramientas útiles para la manipulación de datos, reacomodo, división y combinación de datos, además de esto permite generar estructuras de datos basadas en *NumPy* añadiendo funciones adicionales (McKinney, 2010).

Figura 4.1: Biblioteca de *Pandas*, tomado de The pandas development team, s.f.

***NumPy*,** este paquete es esencial para cualquier proyecto que involucre el cómputo científico (Harris y col., 2020). Es empleada para la generación y manipulación de arreglos n-dimensionales de manera eficiente y sencilla. También incorpora diversos objetos derivados de dicho arreglos y funciones que permiten ejecutar operaciones sobre estos de manera rápida (Harris y col., 2020). Se usa en conjunto con Pandas para la manipulación de los datos de entrada en este trabajo y para las operaciones efectuadas por los distintos algoritmos usados en la predicción de propiedades PVT.

Figura 4.2: Biblioteca de *NumPy*, tomado de The Numpy development team, s.f.

***Matplotlib*,** esta biblioteca posee funciones que facilitan la creación de todo tipo de gráficos con un amplio grado de personalización. Usada generalmente en conjunto con *NumPy*, es la biblioteca empleada en este trabajo para analizar el comportamiento de los datos y los resultados.

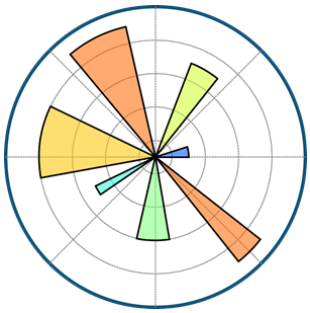


Figura 4.3: Biblioteca de *Matplotlib*, tomado de Yu, s.f.

***Scikit-learn*,** una biblioteca enfocada en ML, optimizada para tener un gran rendimiento. De acuerdo a su página oficial ofrece herramientas sencillas y eficientes para el análisis predictivo de datos, basada en *NumPy* y *SciPy*. Incorpora diversas funciones y algoritmos de ML, enfocados tanto a problemas de clasificación como de regresión y *clustering*, entre los algoritmos que incorpora se incluyen:

1. *Support Vector Machine* (SVM)
2. *Random Forest*
3. Perceptrón multi capas
4. *Nearest Neighbors*
5. *Naive-Bayes*

Además de esto ofrece funciones para el preprocesamiento de datos, descorrelación de variables, generación de conjuntos de datos, conjuntos de datos precargados, descomposición de matrices (como PCA), entre muchas otras. Esta es la biblioteca principal usada para la generación de los modelos descritos en este trabajo.

Figura 4.4: Biblioteca de *Scikit-Learn*, tomado de The scikit-learn development team, s.f.

Cada biblioteca se actualiza constantemente, ya que los desarrolladores siempre corrigen errores e incorporan nuevas funciones, por esto es importante señalar la versión con la que se está trabajando. En la tabla 4.1 se muestran las versiones de cada paquete empleadas en este trabajo, así como la versión de Python.

Tabla 4.1: Versiones del software empleado en este trabajo

Lenguaje		Versión
Python		1.3.4
Paquete	*Pandas*	1.3.4
	NumPy	1.21.5
	Matplotlib	3.5.1
	Scikit-learn	1.0.2

4.2. Modelos de *Machine Learning*

Como se ha mencionado previamente, cada algoritmo de ML tiene un funcionamiento distinto, lo que hará que su desempeño sea diferente incluso para algoritmos de la misma familia con formulaciones matemáticas similares. El comportamiento de la propiedad que se desea predecir, la linealidad de las variables y el conjunto de datos en sí mismo son factores que afectan al algoritmo. Cada

problema debe ser analizado cuidadosamente y de manera independiente para generar el modelo de ML que mejor se adapte a este.

Los valores de las propiedades a predecir son continuas, por lo tanto, el tipo de algoritmos a aplicar deben ser compatibles con este tipo de problemas. Para este trabajo se seleccionaron 6 algoritmos de regresión diferentes y se evaluó su desempeño al predecir p_b y B_{ob}. Los algoritmos se resumen a continuación:

1. *Extremely Randomized Trees* (*Extra Trees*)
2. *Random Forest*
3. Máquina de vectores de soporte ν (*Support Vector Machine*, SVR)
4. Red neuronal artificial (*Artificial Neural Network*, ANN)
5. *K-Nearest Neighbors* (KNN)
6. *Radius Neighbors*

Los primeros dos algoritmos son una variación de árboles de decisión, el tercer algoritmo pertenece a la familia de máquinas de vectores de soporte y el último algoritmo es una variación de KNN. La descripción de cada uno de estos algoritmos se encuentra en el capítulo 3 de este trabajo.

4.3. Datos

La parte medular de cualquier algoritmo de ML se encuentra en los datos que alimentarán al modelo, de esto depende en gran medida el desempeño del algoritmo y la precisión de sus resultados. Esta etapa es quizás una de las más extensas ya que se deben recopilar datos reales que sean representativos del problema a resolver. Además, los datos deben de revisarse minuciosamente para detectar cualquier error, dato faltante o repetido. Tener un buen conjunto de datos es el primer paso para generar un buen modelo predictivo de ML.

4.3.1. Adquisición

La recopilación de datos se efectuó tomando como base diversos artículos de la literatura que presentan datos con las principales propiedades PVT. De estos artículos se generó un conjunto de datos que presenta información de pozos localizados en diferentes partes del mundo, al emplear datos que no se limitan solo a pozos de una cierta región, el algoritmo logra una mejor generalización de los datos y aumenta su rango de aplicación.

Del artículo publicado por Glasø, 1980, se tomaron 45 datos provenientes de pozos del Mar del Norte, Medio Oriente, Estados Unidos y Argelia.

Se extrajeron 160 datos del trabajo publicado por Al-Marhoun, 1988, estos datos PVT pertenecen a 69 muestras de fluidos de 69 pozos de aceite del Medio Oriente.

Del trabajo de Omar y Todd, 1993, se usaron 93 datos recopilados de campos de aceite costa afuera en Malasia.

Se tomaron 195 datos del trabajo de De Ghetto y Villa, 1994, que comprende muestras PVT de fluidos de la cuenca del Mediterráneo, África, el Mar del Norte y el Golfo Pérsico.

Del artículo publicado por BC Gharbi y Elsharkawy, 1999, se tomaron 22 datos provenientes de campos del Medio Oriente.

Con esto se generó un set de 515 datos pertenecientes a campos de distintas regiones del mundo. En la figura 4.5 se muestra un gráfico de pastel con la aportación de cada fuente para el set de datos, los datos de De Ghetto y Villa, 1994 y Al-Marhoun, 1988 son los que tienen la mayor contribución, seguidos por Omar y Todd, 1993, Glasø, 1980 y BC Gharbi y Elsharkawy, 1999.

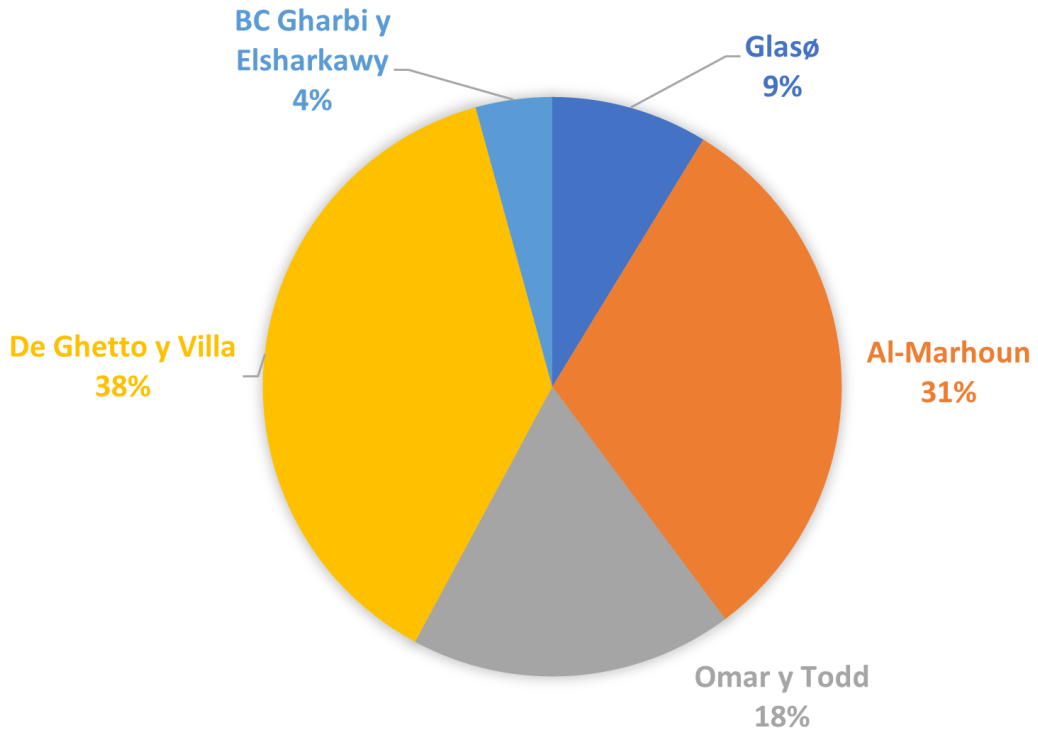


Figura 4.5: Distribución de los datos recopilados

4.3.2. Filtrado

En cualquier aplicación de ML es necesario realizar un filtrado de los datos originales para garantizar los resultados del algoritmo, ya que, como se ha mencionado con anterioridad, la calidad de los datos de entrada es un factor importante para la generalización adecuada del modelo.

Se realizó un filtrado para eliminar cualquier duplicado en los datos, adicionalmente a esto se verificó que todos los datos se encontraran completos y se descartaron aquellos incompletos. Finalmente, se les dio a todos los datos un mismo formato y número de posiciones decimales, esto para darle al algoritmo datos consistentes. Una vez filtrados aquellos datos repetidos o incompletos se obtuvo un set de 509 datos.

El set de datos comprende temperatura de yacimiento, densidad API, densidad relativa del gas, y presión de burbuja; 62 % de los datos presentan el factor de volumen de formación del aceite registrado a la presión de burbuja, el 38 % restante no presenta este dato, el 35 % registra la relación de solubilidad a la presión de burbuja y el 65 % restante no incluye esta propiedad. En la figura 4.6 se muestra gráficamente la proporción de datos faltantes.

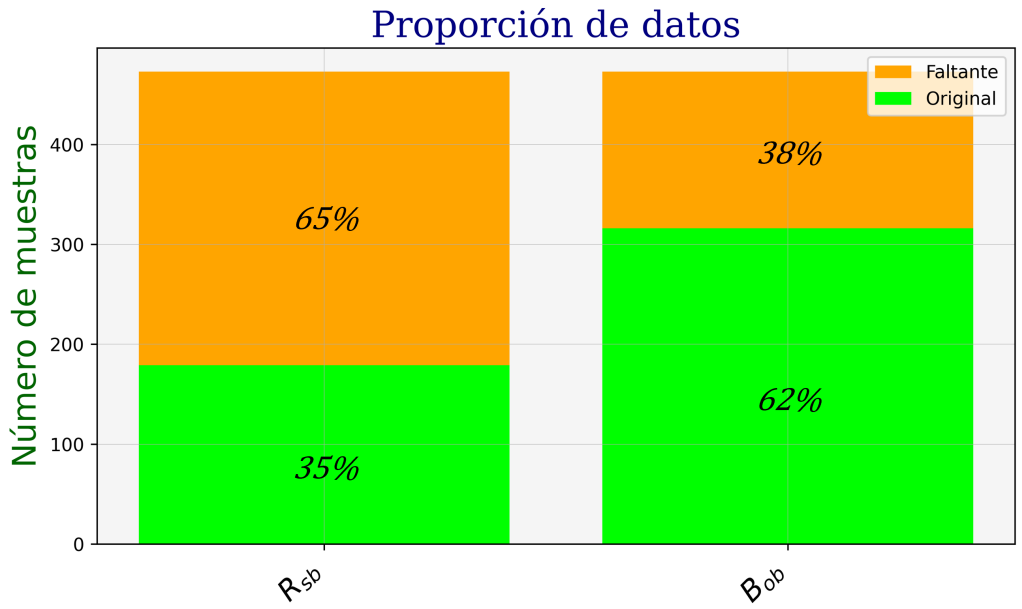


Figura 4.6: Proporción de datos faltantes

Con el fin de generar un conjunto de datos más completo, se propone generar los valores de las magnitudes que no son incluidas en todos los datos, es decir, relación de solubilidad a la presión de burbuja y factor de volumen de formación a la presión de burbuja.

4.3.3. Preprocesamiento

Relación de solubilidad a la presión de burbuja, R_{sb}

Como se mencionó, el 65 % del set no posee el dato de relación de solubilidad a la presión de burbuja. Para poder incluir esta variable como dato de entrada en el algoritmo es necesario obtenerla de alguna manera. Usualmente cuando una propiedad no es determinada en laboratorio se usan correlaciones empíricas para aproximar su valor, la literatura ofrece una gran cantidad de correlaciones que permiten aproximar el valor de esta propiedad como se mostró en un capítulo previo. Para este trabajo se optó por emplear la correlación propuesta por M. B. Standing, 1977 (véase ecuación 4.1), porque ésta permite obtener la relación de solubilidad con las propiedades disponibles (densidad API, densidad relativa del gas, temperatura y presión), en segundo lugar, la mayor parte de los datos cae dentro del rango de aplicabilidad de la correlación. A partir de este punto, cuando se mencione la relación de solubilidad se da por entendido que se está hablando de esta propiedad en el punto de burbuja y se usará indistintamente R_s y R_{sb} para referirse a esta propiedad.

$$R_{sb} = \gamma_g \left[\left(\frac{p}{12.2} + 1.4 \right) 10^{0.0125^\circ API - 0.00091T} \right]^{1.2048} \tag{4.1}$$

Donde:

R_s :SCF/STB

γ_g :1

p :$psia$

T :$^\circ F$

En la figura 4.7 se observa la distribución de los datos originales y los datos generados de R_s, considerando que del total de datos, hubo algunos que fueron eliminados ya que no caen dentro del rango de aplicabilidad de la correlación, por lo que no son de utilidad llegados a este punto. Al analizar la figura 4.7 se puede notar que los datos generados no presentan valores atípicos ya que la distribución de los puntos es bastante similar a la correspondiente a los valores de R_s originales en el set de datos. Cabe resaltar que la mayor parte de los puntos se encuentran en el intervalo de 0 a 1,500 SCF/STB y densidades API de 20 hasta 50 grados.

Factor de volumen de formación del aceite a la presión de burbuja, B_{ob}

Debido a que la mayoría de las correlaciones de la literatura incorporan a la relación de solubilidad como dato de entrada para el cálculo del factor de volumen de formación del aceite, se optó por emplear la correlación propuesta por M. B. Standing, 1977 con la cual se tiene el mismo rango de aplicabilidad de la correlación. A partir de este punto, cuando se mencione el factor de volumen de formación del aceite se da por entendido que se está hablando de esta propiedad en el punto de burbuja y se usará indistintamente B_o y B_{ob} para referirse a esta propiedad.

$$B_o = 0.9759 + 0.00012 \left[R_S \left(\frac{\gamma_g}{\gamma_o} \right)^{0.5} \right]^{1.2} \tag{4.2}$$

Donde:

B_o :rb/STB

γ_o :1

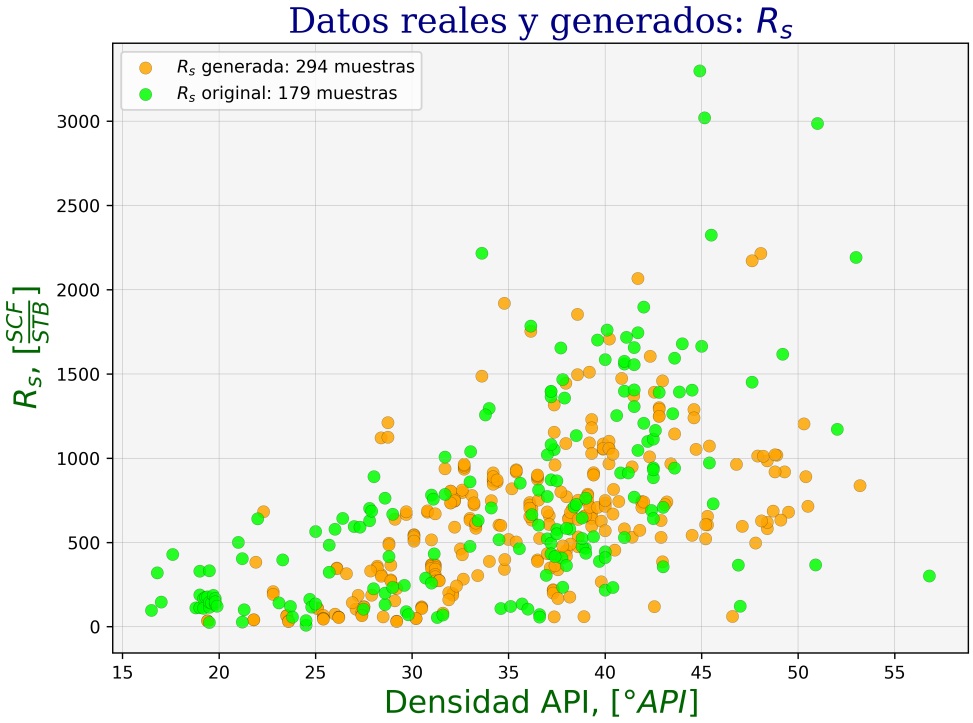


Figura 4.7: Datos reales y generados: R_s

La distribución de los datos de B_{ob} se puede apreciar en la figura 4.8, de manera similar al caso de R_s, se filtraron aquellos datos donde los valores se encontraran fuera del rango de aplicabilidad de la correlación. En esta figura se aprecia que la distribución de los datos generados es similar a la de los datos ya existentes de B_{ob}. Además, la mayor parte de los puntos se encuentran distribuidos desde valores cercanos a 1.0 hasta aproximadamente 2.0 rb/STB y con densidades API que van de 20 a 50 grados.

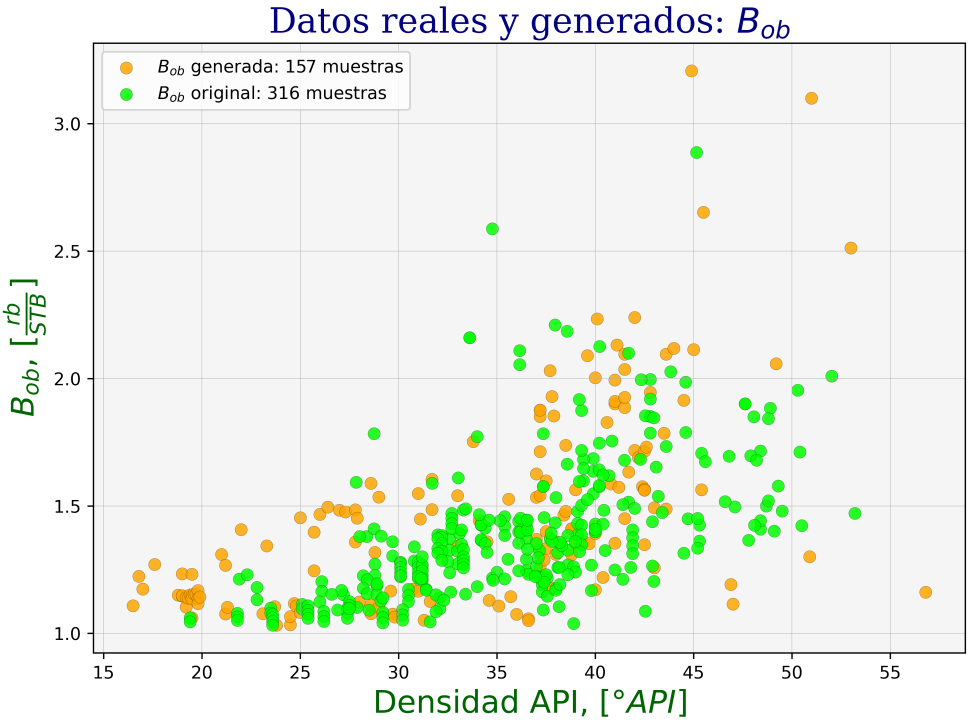


Figura 4.8: Datos reales y generados: B_{ob}

Distribución de datos

Empleando correlaciones se logró generar la relación de solubilidad y el factor de volumen de formación para los datos dentro del rango de aplicabilidad. Al final de esta etapa el set de datos resultante se compone de 473 datos con 6 propiedades (° API , γ_g , T, R_s, B_{ob} y p_b). En la figura 4.9 se puede observar de manera esquemática el proceso descrito.

Adicionalmente, se obtuvo un conjunto de 26 datos pertenecientes a campos nacionales, mismos que son presentados en el apéndice D. Estos datos fueron filtrados adecuadamente y registran las 6 variables de las que ya se ha hablado por lo que no fue necesario ningún tipo de preprocesamiento. Estos datos se toman como set de validación de los modelos de ML en conjunto con algunos datos que serán sustraídos del set generado hasta este momento. Esto es, 39 datos de los 473 y 26 datos de campos mexicanos, se emplearán como set de datos de validación. Con esto el set de datos de entrada para entrenar a los distintos algoritmos de ML queda con 434 datos y el set de validación con 65 datos de los cuales, 26 datos pertenecen a yacimientos de la región del Golfo de México. En la figura 4.10, se esquematiza la proporción de cada conjunto de datos.

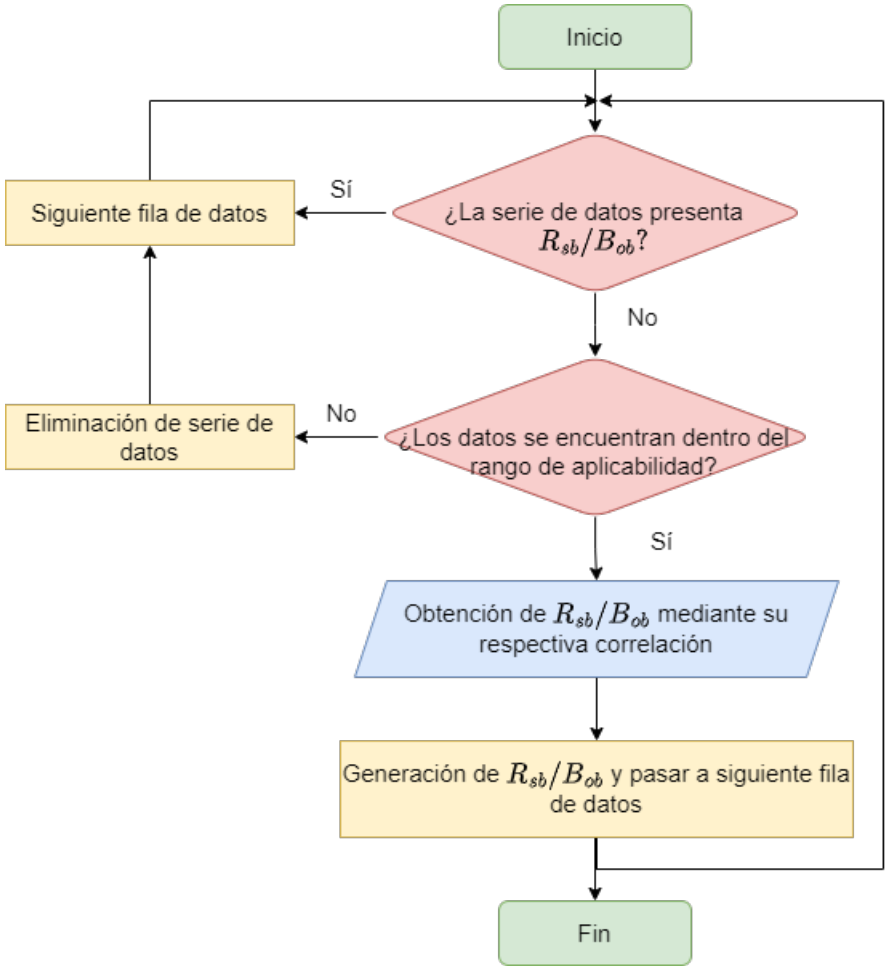


Figura 4.9: Diagrama de flujo para la obtención de R_{sb} y B_{ob}

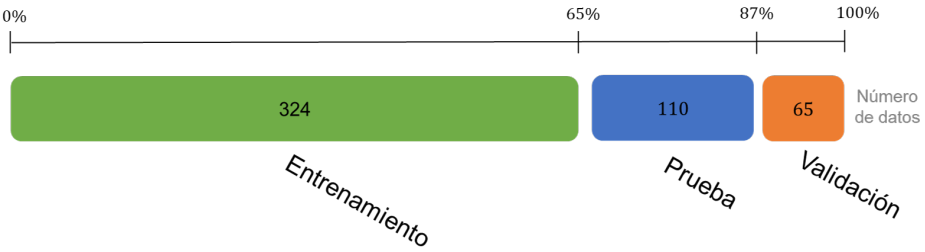


Figura 4.10: Proporción de cada conjunto de datos

Derivado de este procedimiento, se obtuvo un set de datos con 6 propiedades (densidad API ($^{\circ}API$), densidad relativa del gas (γ_g), temperatura de yacimiento (T), relación de solubilidad a la presión de burbuja (R_{sb}), presión de burbuja (p_b) y el factor de volumen de formación a la presión de burbuja (B_{ob})) con 434 valores para cada propiedad. En la tabla 4.2 se muestra una descripción estadística del set de datos de entrada completo. De manera similar en la tabla 4.3 se presenta la descripción estadística para el set de datos de validación, en ambas se muestra la media aritmética (μ), desviación estándar ($d.e.$) y los valores mínimo (min) y máximo (max) para las cuatro variables.

Tabla 4.2: Descripción estadística del set de datos de entrada,
este será empleado para la fase de entrenamiento y prueba

Parámetro	°API	T, °F	γ_g, 1	R_s, SCF/STB	p_b, psi	B_{ob}, rb/STB
μ	35.42	180.1	0.9483	670.7	2246	1.4112
d.e.	7.42	54.0	0.1908	483.2	1413	0.3043
min	16.80	23.0	0.6120	27.8	130	1.0318
max	56.80	341.6	1.5170	3020.0	7127	3.1003

La distribución de datos de la relación de solubilidad y el factor de volumen de formación del aceite ya fueron mostradas en las figuras 4.7 y 4.8. Para visualizar la distribución de las propiedades faltantes, se presentan las figuras 4.11, 4.12 y 4.13 para la temperatura, densidad relativa del gas y presión de burbuja, respectivamente. La mayoría de los datos provienen de yacimientos con temperaturas de 100 a 300 $^{\circ}$ F (véase figura 4.11), la densidad relativa del gas se encuentra en valores que van desde 0.6 hasta 1.4 (véase figura 4.12) y, finalmente se observa en la figura 4.13 que los datos de presión de burbuja se encuentran en el área de 1,000 a 4,000 psi.

Tabla 4.3: Descripción estadística del set de datos de validación
este será empleado para la evaluación de resultados

Parámetro	°API	T, °F	γ_g, 1	R_s, SCF/STB	p_b, psi	B_{ob}, rb/STB
μ	22.68	216.3	1.1102	474.6	1861	1.3354
d.e.	13.07	50.2	0.2171	617.7	1613	0.3885
min	6.00	114.8	0.6630	8.6	107	1.0000
max	53.00	325.4	1.7890	3298.7	6614	3.2076

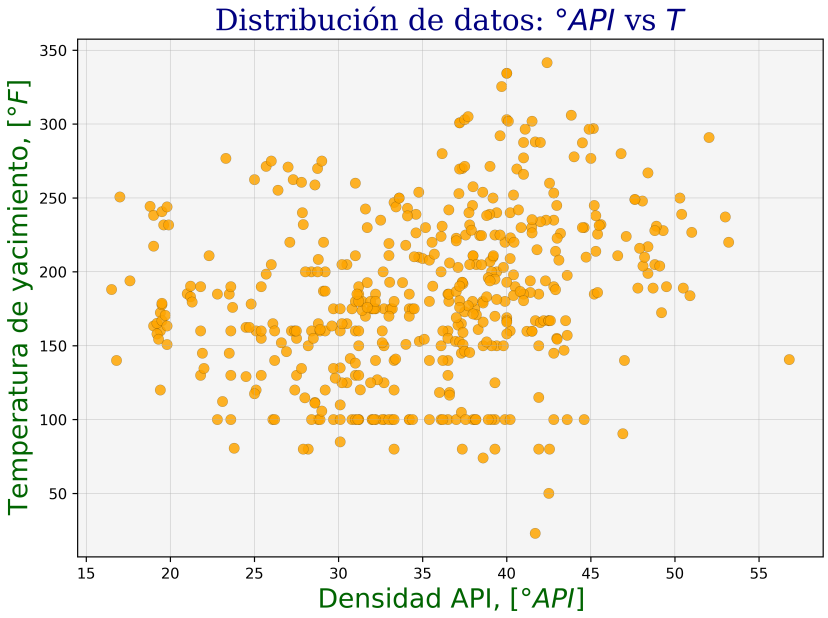


Figura 4.11: Distribución de datos: ° API vs T

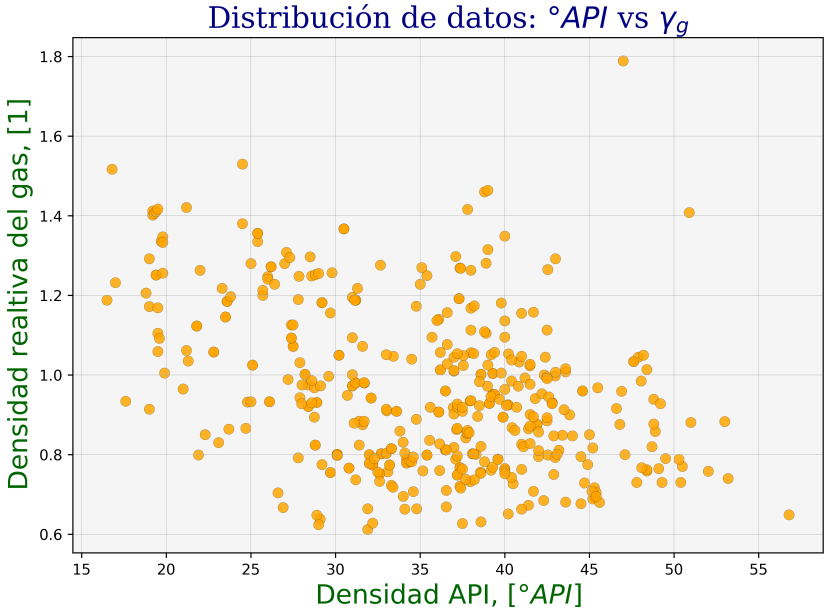


Figura 4.12: Distribución de datos: ° API vs γ_g

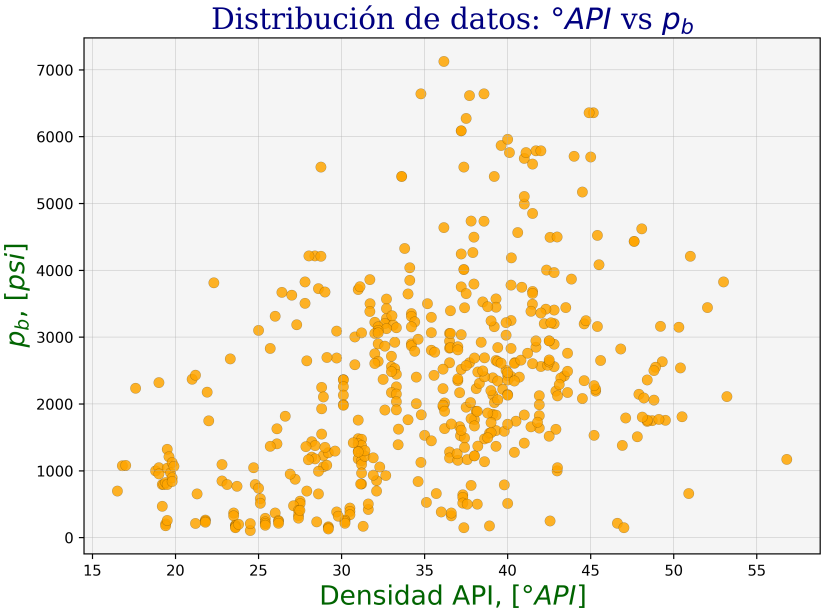


Figura 4.13: Distribución de datos: ° API vs p_b

4.4. Predicción de la presión de burbuja y el factor de volumen de formación del aceite a la presión de burbuja

Con base en el estado del arte y una investigación previamente realizada, las propiedades a determinar en este trabajo son la presión de burbuja y el factor del volumen de formación del aceite a la presión de burbuja. Cabe recalcar que estas propiedades son de gran importancia para la correcta caracterización de un yacimiento, además de que son parámetros cruciales para la correcta simulación del comportamiento de un yacimiento a lo largo de su vida productiva.

Poniendo especial atención en la presión de burbuja, esta es una propiedad que es deseable conocer desde el inicio del desarrollo de un yacimiento ya que marca la presión por debajo de la cual comenzará la manifestación de la fase gas y se tendrá un flujo bifásico que usualmente es más complejo de describir matemáticamente por lo que es recomendable mantener la producción en una presión por encima del punto de burbuja.

4.4.1. Preprocesamiento de datos

En la sección anterior se detalló el proceso llevado a cabo para filtrar el set de datos, por lo qué, para este punto, los datos ya fueron filtrados, sin embargo, aún es necesario evaluar si es necesario realizar alguna transformación (preprocesamiento) en el set de datos de entrada.

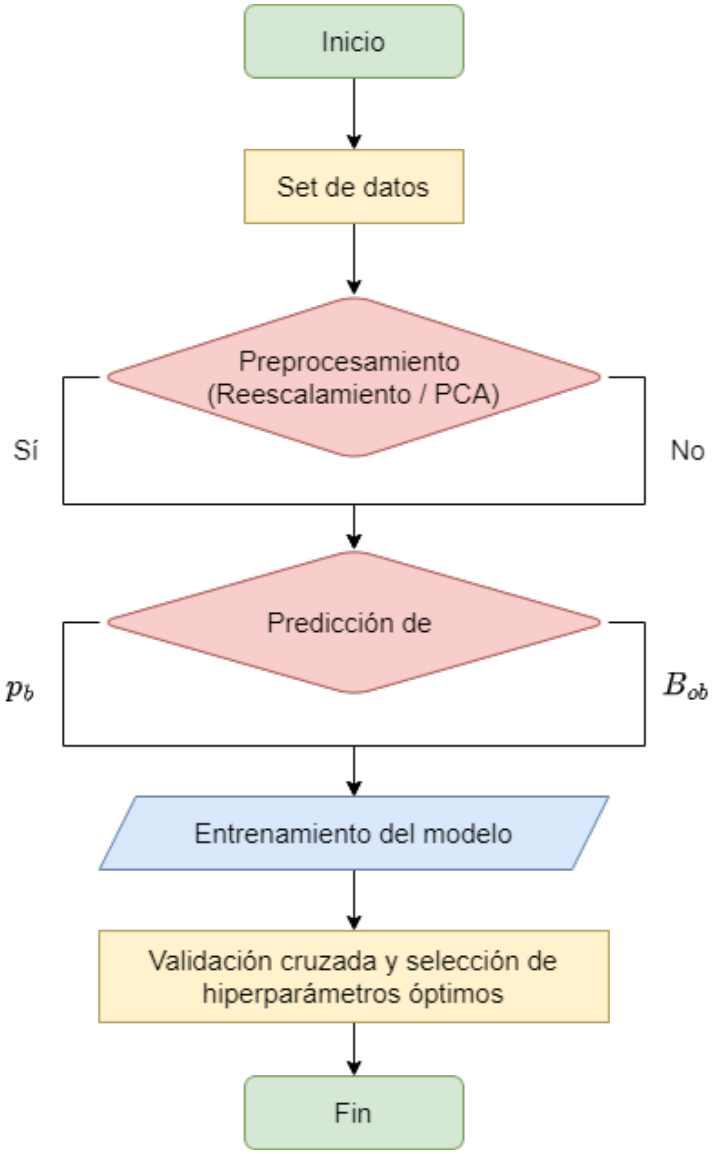


Figura 4.14: Diagrama de flujo del entrenamiento de cada algoritmo de Machine Learning

El preprocesamiento se llevó a cabo de manera separada para cada uno de los seis algoritmos y se evaluó el desempeño de cada modelo transformando los datos y sin haber realizado ninguna transformación. Esta etapa consta de dos transformaciones, una reescalamiento o estandarización de los datos y una transformación PCA. En la figura 4.14 se muestra un esquema del procedimiento seguido para el entrenamiento de cada algoritmo, primero se eligió el set de datos con el que se trabajó y se determinó la variable objetivo (B_{ob} o p_b). Finalmente, se entrenó el modelo y se realizó

una búsqueda de malla para determinar los parámetros del algoritmo y la combinación de las dos transformaciones propuestas que arrojen los mejores resultados, una vez finalizado esto, se tuvo un modelo entrenado con sus híper parámetros ajustados de manera óptima para predecir la variable objetivo seleccionada.

4.4.2. Aplicación de los modelos

Como se mencionó anteriormente, cada modelo se evaluó por separado, de manera similar, se aplicó de manera independiente para la predicción de p_b y B_{ob}. Para esto se plantearon tres casos distintos, uno para p_b y dos para B_{ob}, con el fin de evaluar la capacidad del ML en relación a la predicción de las dos propiedades PVT ya mencionadas:

1. Predicción de p_b
2. Predicción de B_{ob}
3. Predicción de B_{ob} incorporando las predicciones realizadas para p_b

Se aplicaron estos tres escenarios en cada uno de los seis algoritmos, evaluando en cada uno sí era necesario realizar alguna transformación previa como se muestra en la figura 4.14. El diagrama de flujo del procedimiento general seguido en este trabajo presenta cada fase del procedimiento, abarcando desde la adquisición del set de datos, el filtrado, generación de datos y preprocesamiento de los mismos hasta llegar al entrenamiento y evaluación del modelo (véase figura 4.15).

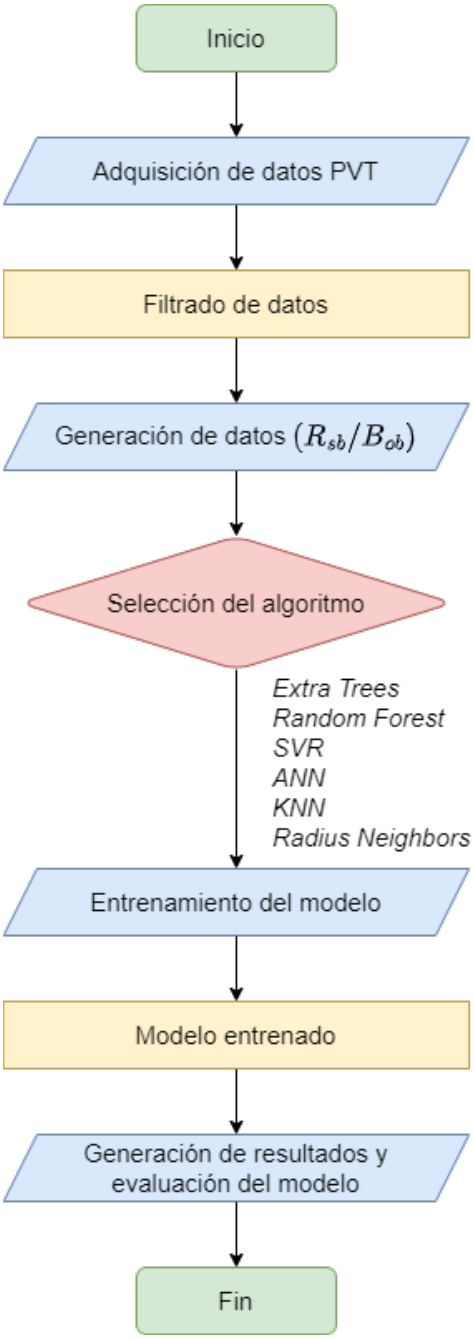


Figura 4.15: Diagrama de flujo de la metodología general para entrenar los modelos de ML

Capítulo 5

Resultados

En este capítulo se muestra el análisis de los resultados obtenidos al aplicar los modelos de Machine Learning (ML) propuestos para la predicción de las variables objetivo siguiendo la metodología descrita en el capítulo anterior. Se realizó una descripción cuantitativa de los resultados con cada modelo sobre el set de datos de prueba, posteriormente se seleccionaron los tres mejores modelos de cada caso y se realizó la evaluación de su desempeño al ser aplicados al set de datos de validación. El capítulo se divide en dos secciones principales, en la primera se analizan los resultados para modelos destinados a predecir la presión de burbuja, y otra donde se detallan los resultados de los algoritmos empleados para a estimar el factor de volumen de formación del aceite a la presión de burbuja; lo descrito a continuación aplica para ambas secciones.

En primer lugar, se mencionan los parámetros internos de cada modelo de ML que mostraron mejor desempeño de acuerdo con la búsqueda de malla realizada. Además, se presentan las transformaciones que se aplicaron a los datos. Asimismo se especifica si no pasaron por ninguna transformación de acuerdo con los resultados de la búsqueda de malla. Posteriormente, se muestra un análisis cualitativo para cada modelo, seguido de esto se muestra una descripción estadística de los datos generados por cada algoritmo.

Adicionalmente, se presenta la misma información para cinco correlaciones (M. B. Standing, 1977, Glasø, 1980, Al-Marhoun, 1988, Dokla y Osman, 1992 y Petrosky y Farshad, 1993) de la literatura, mismas que se presentan detalladamente en el apéndice A. Estas se usan para comparar los resultados obtenidos contra los métodos tradicionales para la predicción de propiedades PVT. Estas cinco correlaciones fueron seleccionadas porque permiten la estimación de las dos variables objetivo con las cuatro propiedades de entrada, es decir, $^{\circ}API$, γ_g, T y R_s; para una mayor claridad, en las figuras y tablas donde aparezcan estas correlaciones, serán abreviadas de la siguiente manera:

1. M. B. Standing, 1977: Standing
2. Glasø, 1980: Glasø
3. Al-Marhoun, 1988: Al-Marhoun
4. Dokla y Osman, 1992: Dokla & Osman
5. Petrosky y Farshad, 1993: Petrosky & Farshad

En la tabla 5.1, se muestra la correlación propuesta por cada autor para aproximar la presión de burbuja y el factor de volumen de formación del aceite a la presión de burbuja. Todas emplean unidades de campo (véase apéndice A para una descripción más detallada de cada una).

Tabla 5.1: Correlaciones empíricas usadas para la comparación de resultados

Correlación	Propiedad	
	p_b	B_{ob}
Standing	$F = \frac{R_{sb}}{\gamma_g}^{0.83} 10^{0.00071T - 0.0125^\circ API}$ $p_b = 18.2\,[F - 1.4]$	$F = R_{sb}\sqrt{\frac{\gamma_g}{\gamma_o}} + 1.25T$ $B_{ob} = 0.9759 + 12 \times 10^{-5} F^{1.2}$
Glasø	$F = \frac{R_{sb}}{\gamma_g}^{0.816} \frac{T^{0.172}}{^\circ API^{0.989}}$ $p_b = 10^{[1.7669 + 1.7447 \log_{10} F - 0.30218(\log_{10} F)^2]}$	$F = R_{sb}\left(\frac{\gamma_g}{\gamma_o}^{0.526}\right) + 0.968T$ $B_{ob} = 1 + 10^{[-6.58511 + 2.91329 log_{10} F - 0.27683(\log_{10} F)^2]}$
Al-Marhoun	$p_b = 5.38088 \times 10^{-3} R_{sb}^{0.715082}\ \gamma_g^{-1.87784}\ \gamma_o^{3.1437}\ (T + 460)^{1.32657}$	$F = R_{sb}^{0.74239}\ \gamma_g^{0.323294}\ \gamma_o^{-1.20204}$ $B_{ob} = 0.497069 + 0.862963 \times 10^{-3}(T + 460) +$ $0.182594 \times 10^{-2} F + 0.318099 \times 10^{-5} F^{-2}$
Dokla & Osman	$p_b = 0.836386 \times 10^4 R_{sb}^{0.724047}\ \gamma_g^{-1.01049}\ \gamma_o^{0.107991}\ (T + 460)^{-0.952584}$	$F = R_{sb}^{0.773572}\ \gamma_g^{0.40402}\ \gamma_o^{-0.882605}$ $B_{ob} = 0.431935 \times 10^{-1} + 0.156667 \times 10^{-2}(T + 460) +$ $0.139775 \times 10^{-2} F + 0.380525 \times 10^{-5} F^{-2}$
Petrosky & Farshad	$F = \frac{R_{sb}^{0.5774}}{\gamma_g^{0.8439}} 10^{[4.561 \times 10^{-5} T^{1.3911} - 7.916 \times 10^{-4}\ ^\circ API^{1.541}]}$ $p_b = 112.727\,[F - 12.34]$	$F = R_{sb}^{0.3738}\left(\frac{\gamma_g^{0.2914}}{\gamma_o^{0.6265}}\right) + 0.24626 T^{0.5371}$ $B_{ob} = 1.0113 + 7.2046 \times 10^{-5} F^{3.0936}$

A continuación, se analiza el error absoluto y relativo de cada modelo, mostrando una descripción estadística de ambos parámetros. Este análisis cuantitativo se presenta para los tres modelos así como para las cinco correlaciones con fines de comparación, de igual manera, se analiza el coeficiente de correlación (R^2) para cada caso, seguido de esto, se muestran gráficos de error absoluto y error relativo para cada modelo y para las dos correlaciones que mostraron un mejor desempeño. Por último, se comparan gráficamente los datos generados por el modelo de ML con el mejor y el peor desempeño contra la correlación con la evaluación más alta, respectivamente.

5.1. Modelos

El set de datos de entrada se separó en tres conjuntos, 65 % de los datos se usaron para el entrenamiento de los seis algoritmos, 22 % de datos fueron destinados al subconjunto de prueba y el 13 % restante compone el subconjunto de validación.

Se evaluaron seis algoritmos distintos de ML para tres escenarios con dos variantes cada uno, los algoritmos fueron evaluados inicialmente con el set de datos de prueba. En la tabla 5.2, se presenta el coeficiente de correlación registrado para cada caso,el cual se generó después de hallar la mejor combinación de hiperparámetros y transformaciones para cada algoritmo con base en una búsqueda

de malla; este valor (R^2) será el que marcará la pauta para la elección de los tres mejores algoritmos en cada caso, mismos que se aplicarán sobre el conjunto de datos de validación para su evaluación.

Tabla 5.2: Coeficiente de correlación (R^2) de los 6 algoritmos para el set de datos de prueba

Modelos	Coeficiente de correlación, %		
	p_b	B_{ob}	$p_b \rightarrow B_{ob}$
Extra Trees	96.97	96.45	96.33
Random Forest	96.65	91.54	94.92
SVR	13.04	96.41	96.93
ANN	86.22	93.11	91.40
KNN	95.13	87.76	92.21
Radius Neighbors	1.45	14.06	18.37

De la tabla 5.2, se puede observar que para estimar p_b, los tres mejores algoritmos son: *Extra Trees*, *Random Forest*, y *KNN*, el algoritmo de *ANN* se mantiene con un desempeño del 86 %, sin embargo, los algoritmos de *SVR* y *Radius Neighbors* muestran una capacidad para ajustarse a p_b muy por debajo de los valores aceptables (70 %), ya que, registran valores de R^2 del 13 y 1.4 %, respectivamente. Para la predicción de B_{ob}, los tres mejores algoritmos son: *Extra Trees*, *SVR* y *ANN*, con valores de R^2 por encima al 95 % para los dos primeros y de 93 % para el tercero, los modelos de *Random Forest* y *KNN* se mantienen con valores aceptables (91 y 87 %, respectivamente) y, finalmente el algoritmo de *Radius Neighbors*, registra un valor que apenas alcanza el 14 %. Para el último caso, los tres mejores algoritmos que lograron predecir B_{ob} a partir de la estimación de p_b ($p_b \rightarrow B_{ob}$), son: *SVR*, *Extra Trees* y *Random Forest*, en cada caso, registran valores de R^2 por encima de 94 %, los algoritmos de *KNN* y *ANN* mantienen una buena precisión al tener un valor superior al 90 %, en último lugar al igual que los dos casos anteriores, se tiene al algoritmo de *Radius Neighbors*.

5.2. Presión de burbuja

En esta sección se evalúan los tres algoritmos (*Extra Trees*, *Random Forest* y *KNN*) con el mejor desempeño para la presión de burbuja y, se lleva a cabo una comparación cualitativa y cuantitativa con respecto a cinco de las correlaciones más usadas disponibles en la literatura.

5.2.1. Parámetros de los modelos

La descripción detallada de cada hiperparámetro y su funcionamiento van más allá del alcance de esta tesis, sin embargo, se menciona la mejor combinación de hiperparámetros derivada de la búsqueda de malla para cada algoritmo. Estos se muestran en la tabla 5.3, para cada modelo aparecen las transformaciones aplicadas a los datos, también se especifica si no pasaron por ninguna transformación, así como los valores óptimos encontrados en la búsqueda de malla para los hiperparámetros particulares de cada algoritmo.

Para el algoritmo de *Extra Trees*, no se necesitó ninguna transformación en los datos de entrada, mientras que para el modelo *Random Forest*, los datos solamente pasaron por una estandarización y finalmente para el modelo *KNN*, los datos se escalaron con el máximo absoluto. En ninguno de los tres casos fue necesaria una transformación PCA. Los dos primeros modelos, que pertenecen a la misma familia de algoritmos, solamente necesitaron de 100 árboles para predecir eficazmente la presión de burbuja. Este es un número reducido de árboles, por lo que el tiempo de cómputo para estos algoritmos no juega un papel relevante, también es notable que el criterio a minimizar en estos dos algoritmos fue el mismo, se trata del error absoluto, mientras que la profundidad de crecimiento máxima para el algoritmo de *Extra Trees* fue mayor que la requerida para el modelo de *Random Forest*. El mínimo de muestras para dividir (hacer una ramificación) y el mínimo de muestras para obtener un nodo hoja fue el mismo para los dos algoritmos (*Extra Trees* y *Random Forest*) de la familia de árboles de decisión. Finalmente, para el último modelo (*KNN*), se aprecia que el número óptimo de vecinos para realizar las predicciones de un punto en específico fue de 4, además, el algoritmo interno para realizar una predicción con base en los puntos vecinos fue el denominado como *Ball tree*.

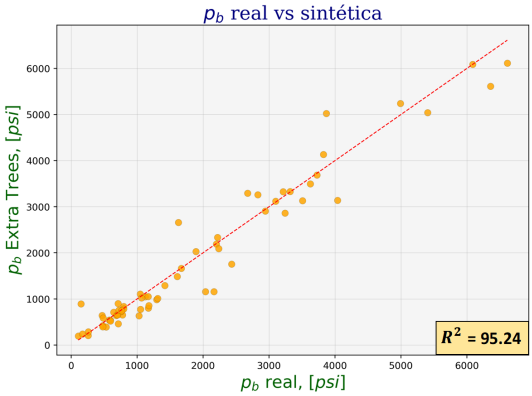
Tabla 5.3: Parámetros de los modelos para p_b

	Parámetro	Valor	
Extra Trees	Escalamiento	Ninguno	Modelo 1
	PCA	No	
	Número de árboles	100	
	Criterio	Error absoluto	
	Profundidad máxima	40	
	Mínimo de muestras para dividir	2	
	Mínimo de muestras para nodo hoja	1	
Random Forest	Escalamiento	Estandarización	Modelo 2
	PCA	No	
	Número de árboles	100	
	Criterio	Error absoluto	
	Profundidad máxima	25	
	Mínimo de muestras para dividir	2	
	Mínimo de muestras para nodo hoja	1	
KNN	Escalamiento	Máximo absoluto	Modelo 3
	PCA	No	
	Algoritmo	Ball tree	
	Número de vecinos	4	

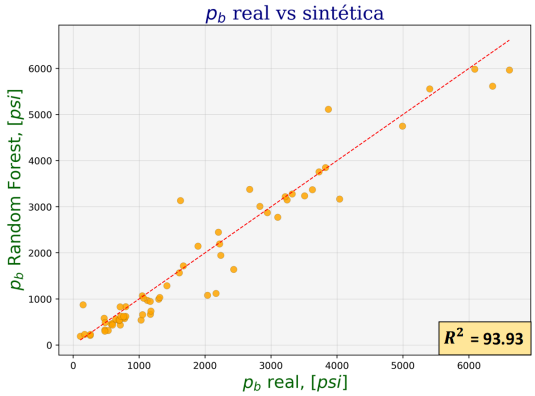
5.2.2. Datos generados

En la figura 5.1, se muestra la comparación en la predicción de los modelos al ser aplicados sobre el conjunto de datos de validación así como el coeficiente de correlación para cada algoritmo.

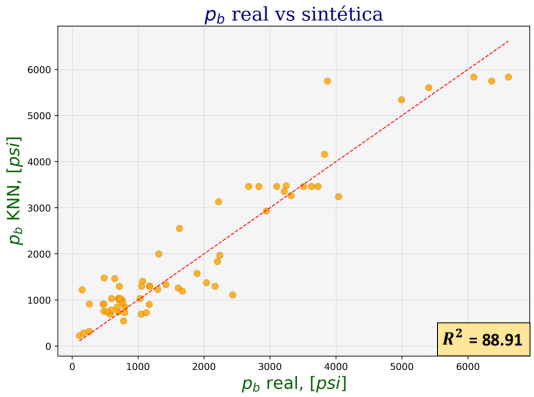
En el eje de las abscisas se presentan los valores reales de p_b, mientras que en el eje de las ordenadas se colocan los valores generados de p_b, cualitativamente se puede notar que entre más se asemejen a la línea identidad (véase línea roja punteada en cada gráfica de la figura 5.1), los puntos tienen mayor calidad en las predicciones, de manera contraria, entre más dispersos se encuentren los puntos con relación a la línea identidad se puede asegurar que la calidad de las predicciones es menor, ya que se encuentran más alejados del valor real.



(a) *Extra Trees*



(b) *Random Forest*



(c) *KNN*

Figura 5.1: p_b real vs sintética

En la figura 5.1a, se observa que el modelo de *Extra Trees* los datos generados (sintéticos) se encuentran cualitativamente más dispersos por encima de los 2,000 psi. La figura 5.1b, perteneciente al modelo de *Random Forest*, muestra una mayor dispersión de los puntos (respecto a la figura 5.1a), especialmente en el rango de los 1,600 a los 4,000 psi. Para el modelo de *KNN* presentado en la figura 5.1c, la dispersión en los puntos es más evidente que en los dos casos anteriores, a lo largo de toda la gráfica los puntos están más alejados de los valores reales, la única zona donde los puntos se acercan más a los valores reales se encuentra por debajo de los 1,000 psi. Cabe resaltar que del los 2,500 a 3,800 psi, los valores predichos registran valores de p_b muy parecidos entre si y con una tendencia aproximadamente constante, ya que su distribución aparenta una línea horizontal.

La tabla 5.4, muestra una descripción estadística de los valores generados para cada modelo, esta incluye la media aritmética (μ), desviación estándar ($d.e.$), y valores mínimo (min) y máximo (max) de los datos de presión de burbuja reales así como aquellos generados mediante los modelos de ML, adicionalmente se incluyen los datos para las cinco correlaciones descritas previamente.

Tabla 5.4: Descripción estadística de p_b real y generada

Modelo		μ, [psi]	d.e., [psi]	min, [psi]	max, [psi]
Real		1861	1613	107	6614
ML	Extra Trees	1899	1615	196	6274
	Random Forest	1858	1594	185	6299
	KNN	1991	1447	410	5746
Correlaciones	Standing	1923	1841	24	9543
	Glasø	2406	1620	11	7877
	Al-Marhoun	1649	1876	39	9390
	Dokla & Osman	1292	1369	59	6745
	Petrosky & Farshad	1571	2107	-1159	9175

Con base en lo registrado en esta tabla se puede ver que el modelo con el valor medio (μ) más similar al valor real es el de *Random Forest* , con tan solo 3 psi de diferencia, seguido por el modelo *Extra Trees* con 38 psi de diferencia; el algoritmo de *KNN* , registra una diferencia mayor a 100 psi. La correlación que se asemeja más a la distribución real es la de M. B. Standing, 1977, con una media similar al modelo de *Extra Trees* aproximadamente, en este sentido, las demás correlaciones tienen una diferencia en el valor medio no menor a 200 psi. Para el valor mínimo, el modelo de *Extra Trees* y *Random Forest* registran valores similares entre si, por otra parte, las correlaciones registran menores diferencias respecto al valor real en este parámetro. Para el valor máximo, la única correlación con una diferencia menor a 1,200 psi respecto al valor real, es la de Dokla y Osman, 1992, con 132 psi; los tres modelos de ML presentan un valor máximo más cercano al real en relación con las correlaciones, en ninguno de los tres casos se tiene una diferencia mayor a 900 psi.

5.2.3. Error relativo, error absoluto y coeficiente de correlación

En esta sección se muestra el error relativo, el error absoluto y el coeficiente de correlación para los tres mejores modelos encargados de predecir la p_b, esto se aplica para las dos variantes en el set de datos de entrada. Estos resultados se reportan de manera cuantitativa (véase tabla 5.5) y cualitativa (véase figuras 5.2, 5.3, 5.4, 5.7, 5.8 y 5.9).

La tabla 5.5, presenta la descripción estadística de los errores relativos, absolutos y coeficientes de correlación reportados por cada modelo y las cinco correlaciones seleccionadas. El coeficiente de correlación se muestra nuevamente con fines de comparación entre los modelos y las correlaciones.

Tabla 5.5: Error y R^2 para los modelos de p_b

Modelo		Error relativo, [%]				Error absoluto, [psi]				R^2, [%]
		μ	d.e.	min	max	μ (MAE)	d.e.	min	max	
ML	Extra Trees	21.04	26.74	0.00	174.74	236	259	0	1272	95.24
	Random Forest	22.33	29.08	0.00	210.94	272	288	0	1362	93.93
	KNN	48.08	62.93	0.81	288.82	410	343	13	1588	88.91
Correlaciones	Standing	28.12	20.51	0.64	102.92	428	495	9	3184	83.44
	Glasø	83.58	91.51	1.52	340.32	829	662	12	2826	56.34
	Al-Marhoun	25.49	23.40	1.65	153.49	493	652	13	3031	74.18
	Dokla & Osman	36.63	21.08	0.23	123.81	626	664	4	2398	67.81
	Petrosky & Farshad	84.40	175.89	0.32	1179.62	621	474	2	2816	76.33

El modelo con la media de error relativo más reducida es el de *Extra Trees*, mostrando un porcentaje de error medio del 21 %. El modelo *Random Forest*, muestra valores similares. Solamente el tercer algoritmo presenta un valor medio de error relativo significativamente distinto, con más del 47 %. Las dos correlaciones con el mejor desempeño para este parámetro son las escritas por M. B. Standing, 1977 y Al-Marhoun, 1988, mostrando valores mayores a los de *Extra Trees* y *Random Forest*. El error relativo máximo es otro parámetro del que vale la pena realizar un análisis más detallado. En este sentido, las correlaciones de M. B. Standing, 1977, Al-Marhoun, 1988 y Dokla y Osman, 1992, muestran valores máximos más reducidos que los modelos de ML, solamente el algoritmo de *Extra Trees* tiene valores aproximadamente similares. En las figuras 5.2, 5.3 y 5.4, se ve reflejado cualitativamente el error relativo y absoluto para los tres modelos.

Ahora, se describirá lo observado respecto a los datos del error absoluto. El modelo con el mínimo error absoluto medio es el de *Extra Trees*, seguido de de *Random Forest* y *KNN*. Además de esto, los valores reportados por los tres modelos superan a las cinco correlaciones, de las cuales, las más destacables son las de M. B. Standing, 1977 y Al-Marhoun, 1988, sin embargo, se encuentran a no menos de 190 psi del mejor modelo de ML (*Extra Trees*). Siguiendo con el análisis de esta tabla, el error absoluto mínimo respecto al valor real es bastante similar en las correlaciones y ligeramente menor para los modelos de ML, respecto a estos últimos, el algoritmo de *Extra Trees* y *Random Forest*, presentan valores mínimos menores a 1 psi. Las cinco correlaciones reportan

valores generalmente más altos para este parámetro, sin embargo, la diferencia no es tan significativa respecto a los algoritmos de ML. Los tres modelos arrojaron errores absolutos máximos menores a los de las cinco correlaciones. En este sentido la correlación con el valor más reducido (Dokla y Osman, 1992) es superior a los modelos de ML una diferencia de 810 psi para el caso de *KNN* y hasta más de 1,100 respecto al mejor modelo (*Extra Trees*). En general las correlaciones presentan errores máximos mayores a 2,000 psi, mientras que los modelos de ML no superan los 1,900 psi.

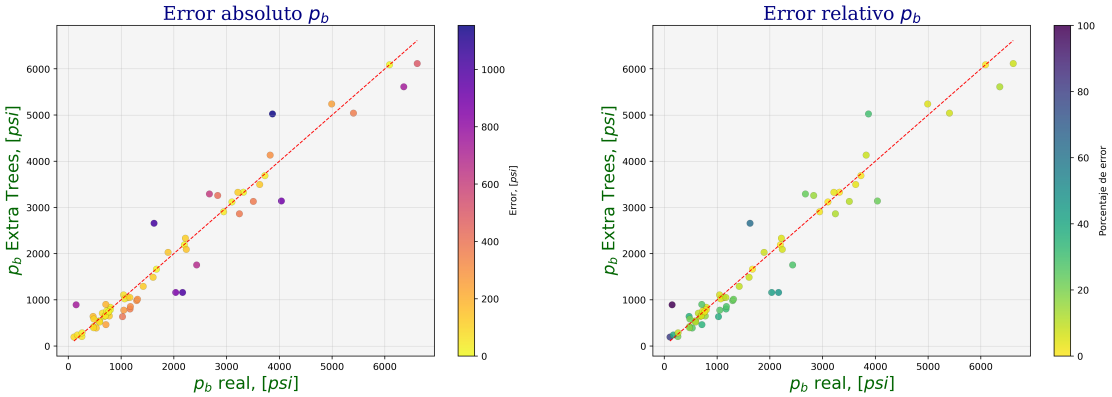


Figura 5.2: Error del modelo *Extra Trees* para p_b

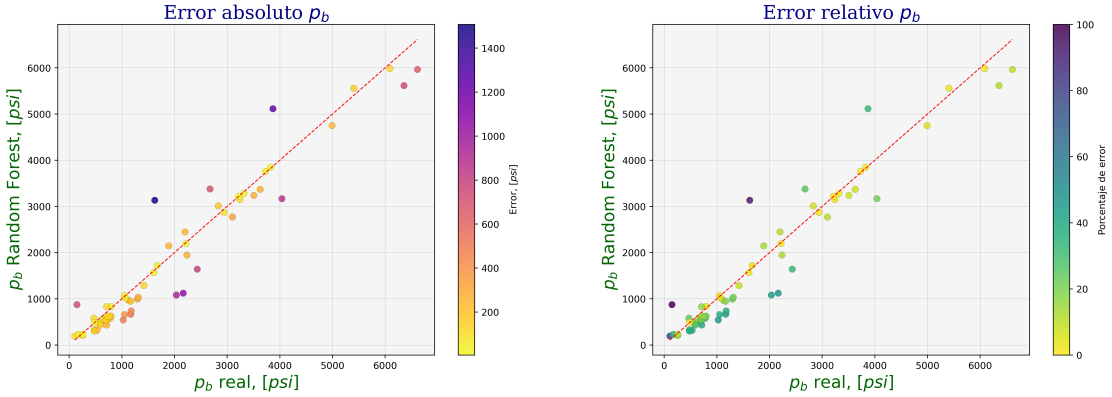


Figura 5.3: Error del modelo *Random Forest* para p_b

Finalmente, se analiza el coeficiente de correlación (R^2) para los tres modelos de ML y las cinco correlaciones. Tanto el modelo de *Extra Trees* como *Random Forest* muestran valores que superan el 90 %, siendo de 95 % y 93 %, respectivamente. Los tres algoritmos (incluso *KNN*, que registra el valor más bajo), superan bajo esta métrica a las cinco correlaciones presentadas. Esto se hace notar debido

a que la correlación con un valor de R^2 más elevado, que es la propuesta por M. B. Standing, 1977, presenta un valor de 83 %, lo que la coloca en un 5.5 % por debajo del peor modelo (*KNN*) y a 11.8 % del modelo con las mejores estimaciones (*Extra Trees*).

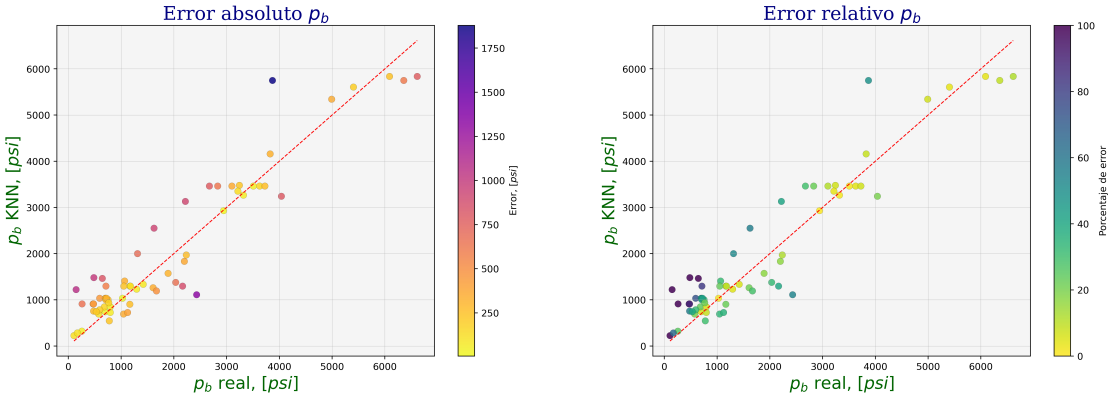


Figura 5.4: Error del modelo *KNN* para p_b

Así como las figuras 5.2, 5.3 y 5.4, se presentan gráficas similares para las dos correlaciones con el coeficiente de correlación más alto, ya que este es el principal indicador para evaluar la precisión de las estimaciones. Estas correlaciones son las propuestas por M. B. Standing, 1977 con un valor de R^2 del 83 % (véase figura 5.5), y por Petrosky y Farshad, 1993 con un valor de R^2 del 76 % (véase figura 5.6), respectivamente.

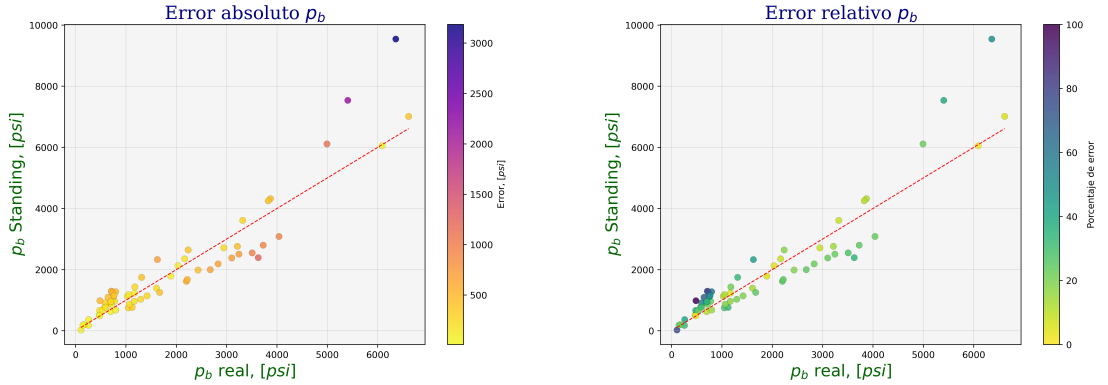


Figura 5.5: Error de la correlación de M. B. Standing, 1977 para p_b

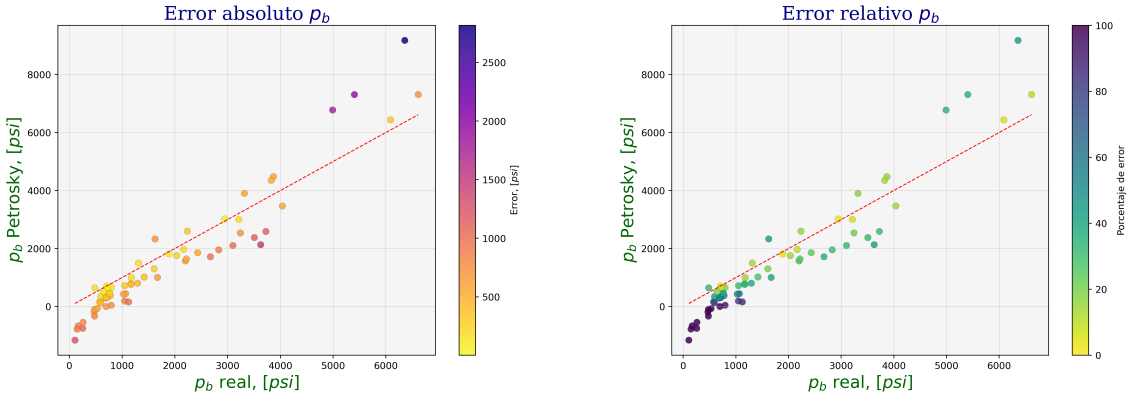


Figura 5.6: Error de la correlación de Petrosky y Farshad, 1993 para p_b

En las figuras 5.7, 5.8 y 5.9, se muestran gráficos de barras con el error relativo medio, el error medio absoluto (MAE) y el coeficiente de correlación (R^2), respectivamente. Esto se presenta para cada modelo y las cinco correlaciones; los valores mostrados en estas tablas se muestran para tener una mejor apreciación de los datos obtenidos en esta tesis.

La figura 5.7, muestra que solamente el modelo de *KNN* presenta un error relativo medio mayor a la media de todos los modelos. De acuerdo con esta figura, los dos algoritmos de la familia de árboles de decisión tienen un desempeño similar y ambos arrojan errores relativos medios menores a las correlaciones. De estas últimas, solamente la presentada por M. B. Standing, 1977 y Al-Marhoun, 1988, tienen valores que se asemejan a los dos modelos de ML de la familia de árboles de decisión.

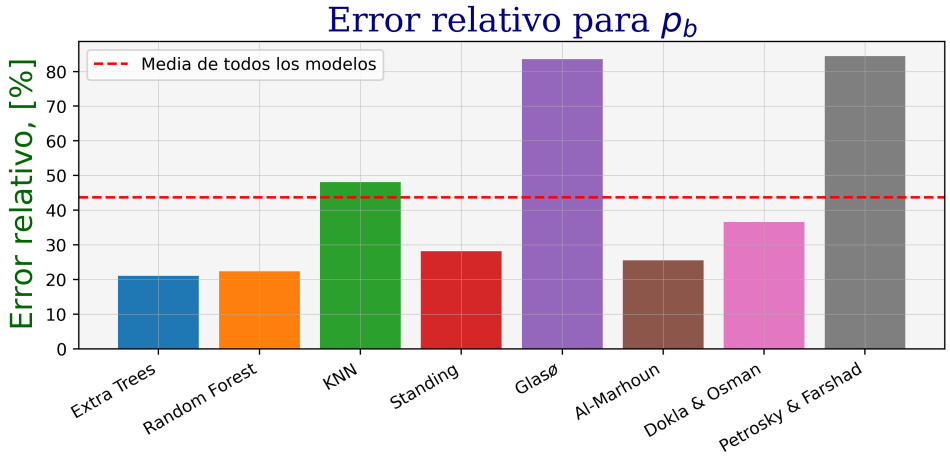


Figura 5.7: Error relativo para p_b

Analizando la figura 5.8, se observa que todos los modelos de ML tienen un MAE por debajo de la media y solo la correlación de M. B. Standing, 1977, se mantiene con un valor de error medio relativamente cercano al de los algoritmos de ML, especialmente del modelo de *KNN*.

En la figura 5.9 se puede corroborar que, los algoritmos de ML tienen una mayor precisión respecto a las correlaciones empíricas tradicionales, con valores significativamente más altos de R^2. De acuerdo con este gráfico de barras, solamente la correlación propuesta por M. B. Standing, 1977, tiene un valor por debajo de la media de todos los modelos. Esto puede deberse a que fue utilizada para generar datos faltantes en la etapa de preprocesamiento en el capítulo anterior.

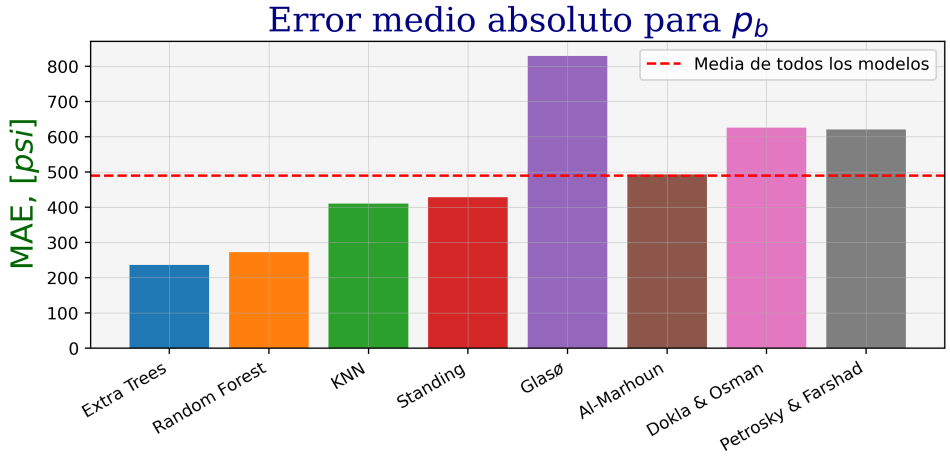


Figura 5.8: Error medio absoluto (MAE) para p_b

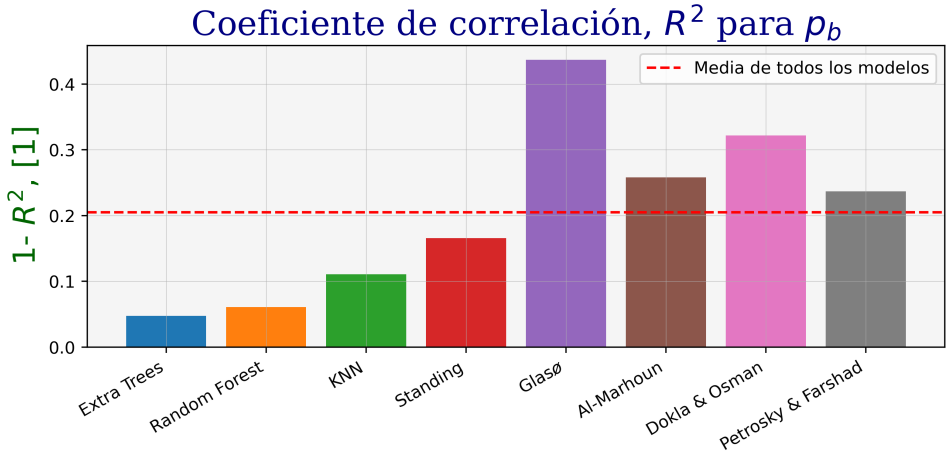


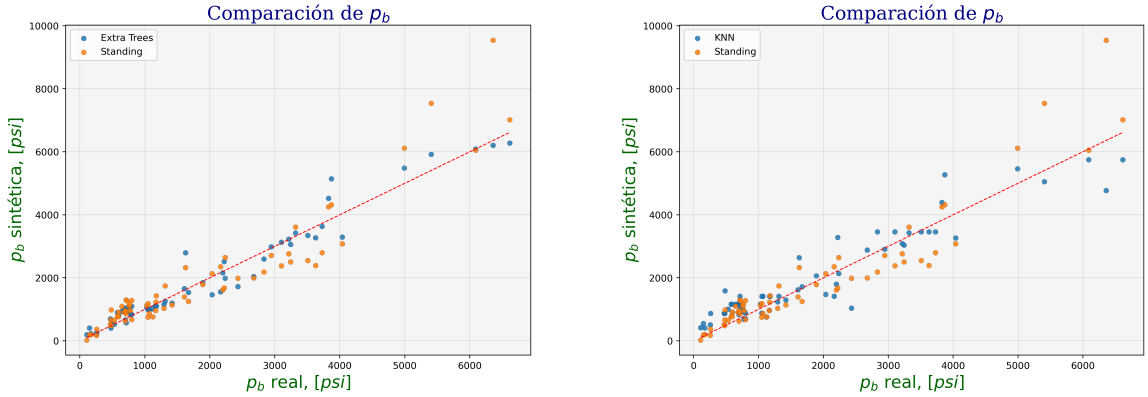
Figura 5.9: Coeficiente de correlación (R^2) para p_b

5.2.4. Comparación con la correlación de M. B. Standing, 1977

Finalmente, para darle otra perspectiva a los resultados obtenidos en esta sección, se presenta en la figura 5.10a una comparación en la distribución de los puntos entre el algoritmo con el mejor valor de R^2, contra la correlación con mejor desempeño, es decir, se presenta el algoritmo de *Extra Trees* contra la correlación de M. B. Standing, 1977. De manera análoga, se muestra una figura similar para comparar al algoritmo con el desempeño más bajo de acuerdo a su valor de R^2. Esto corresponde a *KNN* contra la correlación ya mencionada (véase 5.10b).

Para la figura 5.10a se puede observar que los puntos pertenecientes a la correlación, se encuentran más alejados de la línea identidad, principalmente de los 2,800 a los 4,000 psi, por encima de 5,000 psi, la correlación muestra puntos aún más alejados del valor real. En general, la consistencia del modelo de ML es mayor, ya que los puntos están menos dispersos respecto a la línea roja, manteniendo un comportamiento relativamente similar a lo largo de todo el rango de presión.

Respecto a la figura 5.10b, se nota que los puntos correspondientes al algoritmo de *KNN*, están uniformemente más alejados de la línea roja, sin embargo, hay zonas donde se las predicciones se alejan significativamente, lo que afecta el comportamiento promedio de las predicciones. El modelo de ML muestra una mayor precisión respecto a la correlación arriba de los 5,000 psi.



(a) *Extra Trees* para p_b

(b) *KNN* para p_b

Figura 5.10: Comparación de los modelos de ML vs correlación de M. B. Standing, 1977 para p_b

5.3. Factor de volumen de formación del aceite a la presión de burbuja

Para la segunda parte de este capítulo se presenta una evaluación detallada de los modelos que registraron una mayor precisión al predecir el factor de volumen de formación del aceite a la presión de burbuja, además de presentar la comparación con las correlaciones ya mencionadas.

5.3.1. Parámetros de los modelos

En las tablas 5.6 y 5.7, se resumen los parámetros óptimos (de acuerdo a la búsqueda de malla), de los tres algoritmos con mejores resultados para estimar B_{ob}, así como las transformaciones realizadas sobre los datos previos al entrenamiento del modelo.

Tabla 5.6: Parámetros de los modelos para B_{ob}

	Parámetro	Valor	
Extra Trees	Escalamiento	Ninguno	Modelo 1
	PCA	No	
	Número de árboles	400	
	Criterio	Error absoluto	
	Profundidad máxima	45	
	Mínimo de muestras para dividir	2	
	Mínimo de muestras para nodo hoja	1	
SVR	Escalamiento	Máximo absoluto	Modelo 2
	PCA	No	
	C	3.5	
	nu	0.75	
	Iteraciones máximas	10000	
	kernel	rbf	
	gamma	scale	
	Tolerancia	0.001	
ANN	Escalamiento	Estandarización	Modelo 3
	PCA	Si	
	Función de activación	Identidad	
	Capas ocultas	2	
	Neuronas en la 1era capa oculta	6	
	Neuronas en la 2da capa oculta	6	
	Iteraciones máximas	4000	
	Solver	adam	
	Ritmo de aprendizaje	0.001	
	Tolerancia	0.0001	

Para comenzar, se discute sobre el primer bloque de modelos (véase tabla 5.6), es decir, aquellos que no toman a p_b como dato de entrada. Respecto al modelo de *Extra Trees*, no se requirió ningún tipo de transformación en los datos, se generaron 400 árboles de decisión dentro del algoritmo, al igual que los modelos de p_b, se tomó como criterio a minimizar el error absoluto y la profundidad máxima de crecimiento de los árboles se limitó en un valor de 45. El algoritmo de *SVR*, solamente ocupó una transformación de escalamiento, ésta fue de máximo absoluto, la transformación de PCA no fue necesaria, alcanzó los mejores resultados cuando se fijó el límite de iteraciones del algoritmo en 10,000 y, la tolerancia para detener el algoritmo fue de 0.001. La penalización en el término de error está determinada por el parámetro C, que en este caso es de 3.5. El término nu determina el límite superior de la fracción de errores en el entrenamiento del algoritmo, también determina el límite inferior de la fracción de vectores de soporte (Pedregosa y col., 2011). Finalmente, para el algoritmo de *ANN*, se empleó un escalamiento de estandarización y una transformación PCA sobre los datos de entrada sin eliminar ningún componente. La arquitectura de la red neuronal consistió en una neurona para cada dato de entrada en la capa 1 (capa de entrada), 6 neuronas en la primera y segunda capa oculta y finalmente 1 neurona en la capa final (capa de salida), la función identidad arrojó los mejores resultados como función de activación.

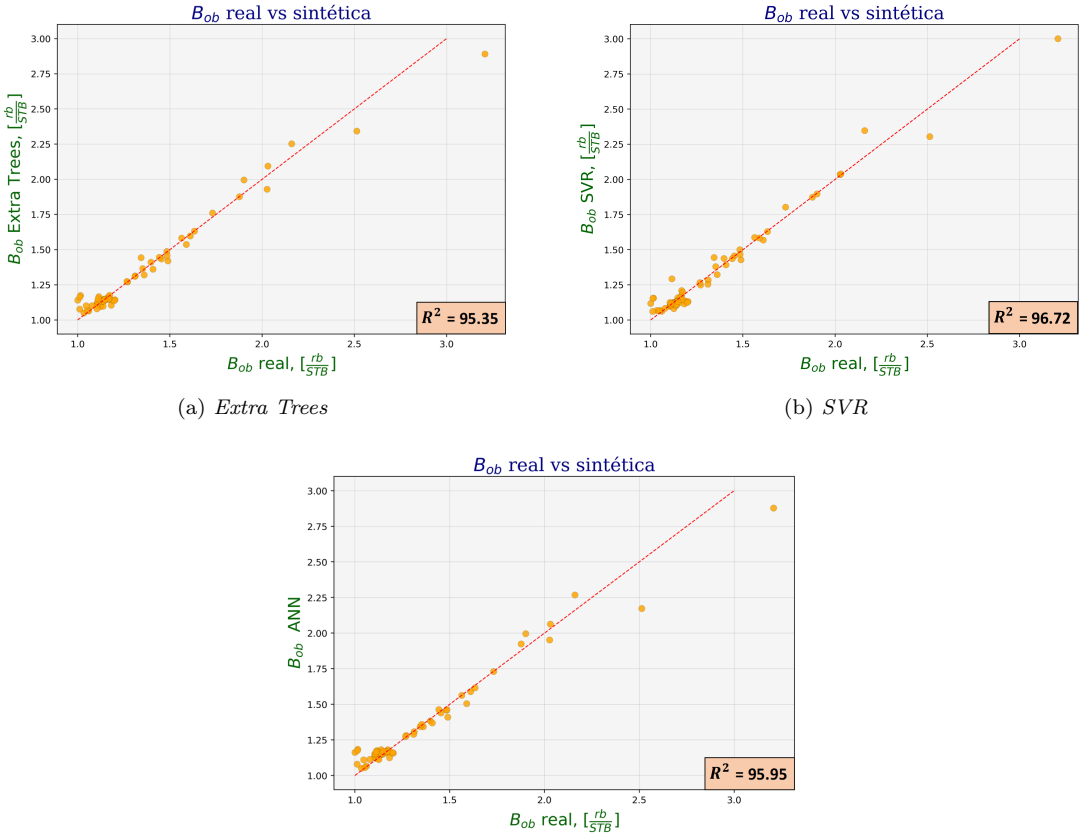
Tabla 5.7: Parámetros de los modelos de $p_b \rightarrow B_{ob}$ para B_{ob}

	Parámetro	Valor	
SVR	Escalamiento	Máximo absoluto	Modelo 1
	PCA	No	
	C	1.5	
	nu	0.5	
	Iteraciones máximas	10000	
	kernel	rbf	
	gamma	scale	
	Tolerancia	0.001	
Extra Trees	Escalamiento	Máximo absoluto	Modelo 2
	PCA	No	
	Número de árboles	100	
	Criterio	Error absoluto	
	Profundidad máxima	40	
	Mínimo de muestras para dividir	2	
	Mínimo de muestras para nodo hoja	1	
Random Forest	Escalamiento	Estandarización	Modelo 3
	PCA	No	
	Número de árboles	100	
	Criterio	Error absoluto	
	Profundidad máxima	30	
	Mínimo de muestras para dividir	2	
	Mínimo de muestras para nodo hoja	1	

Para los modelos del segundo bloque se realizó un análisis similar, los parámetros de estos modelos se ven reflejados en la tabla 5.7. Para el modelo de *SVR*, se empleó un escalamiento de máximo absoluto, se fijó el valor de C igual a 1.5 y nu de 0.5, se fijaron 10,000 iteraciones como máximo. Para el modelo de *Extra Trees*, se empleó un escalamiento máximo absoluto y no fue necesario transformar los datos mediante PCA. Solamente se necesitó entrenar a 100 árboles, limitando su crecimiento hasta un valor de 40 y usando como criterio de optimización, al máximo absoluto. Finalmente, para el modelo de *Random Forest*, se empleó una transformación de estandarización sobre los datos de entrada, el modelo generó 100 árboles de decisión, buscando optimizar el error absoluto y con un valor en el parámetro de profundidad máxima de 30.

5.3.2. Datos generados

La figura 5.11 muestra la comparación de las predicciones hechas por cada modelo de ML contra el valor real de B_{ob}.



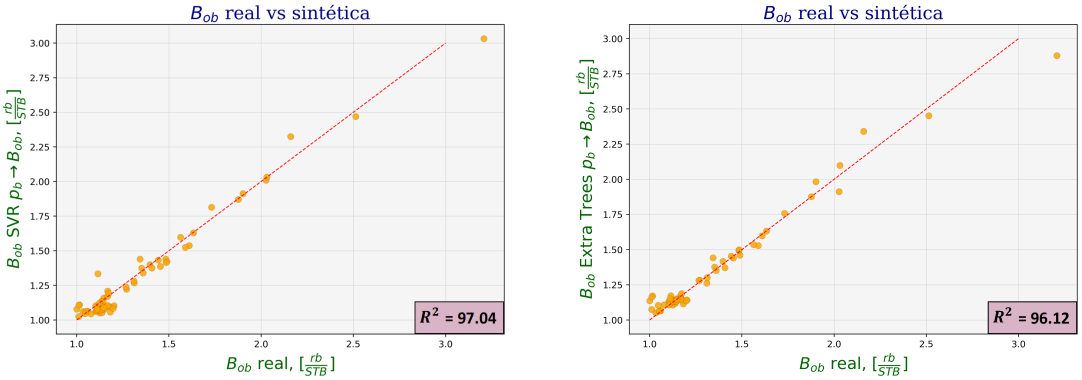
(a) *Extra Trees*

(b) *SVR*

(c) *ANN*

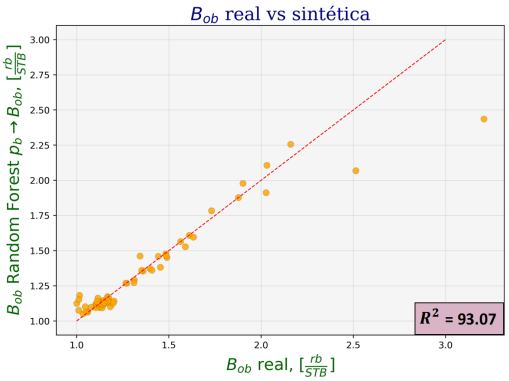
Figura 5.11: B_{ob} real vs sintética

Los datos para el algoritmo de *Extra Trees*, mostrados en la figura 5.11a se puede ver que los puntos se encuentran bastante apilados y muy cerca de la línea identidad, mientras que por encima de los 1.8 rb/STB se pierde la precisión de las estimaciones. La distribución de puntos para el modelo de *SVR* que aparecen en la figura 5.11b, muestran un buen ajuste; por encima de los 2.2 rb/STB se aprecia una divergencia respecto a los valores reales. Finalmente, en la figura 5.11c, se observa una distribución muy similar de los puntos; las predicciones del modelo tienden a disminuir su precisión cuando B_{ob} es mayor a 2.0 rb/STB.



(a) *SVR* $p_b \rightarrow B_{ob}$

(b) *Extra Trees* $p_b \rightarrow B_{ob}$



(c) *Random Forest* $p_b \rightarrow B_{ob}$

Figura 5.12: B_{ob} real vs sintética ($p_b \rightarrow B_{ob}$)

De forma análoga, la figura 5.12 incluye los tres modelos con el mejor desempeño para predecir B_{ob}, tomando en cuenta como dato de entrada la p_b generada con el mismo algoritmo de ML, se usa el símbolo $p_b \rightarrow B_{ob}$ para denotar estos algoritmos. En la figura 5.12a, se muestra el ajuste de los datos con el algoritmo de *SVR*, arriba de los 2.5 rb/STB, los puntos del modelo tienen una mayor distancia de la línea roja. Para el modelo de *Extra Trees*, mostrado en la figura 5.12b, se aprecia una dispersión similar, la diferencia más notoria en la dispersión de los puntos se da por encima de

los 2.0 rb/STB. En tercer, lugar se tiene al algoritmo de *Random Forest*, para este, se puede ver en su figura (véase 5.12c), que el modelo se ajusta bastante bien a los valores reales de B_{ob}, excepto por encima de los 2.2 rb/STB.

En la tabla 5.8 se presenta una descripción estadística de los datos reales y los generados por los diferentes modelos de ML y correlaciones para B_{ob}. Para el primer bloque de algoritmos se tiene que el modelo de ML con la menor diferencia en el valor medio respecto al dato real es el de *ANN*, con una diferencia bastante reducida, alrededor de $4x10^{-3}$ rb/STB. Al comparar contra las correlaciones se nota que las escritas por Al-Marhoun, 1988 y Petrosky y Farshad, 1993 son las únicas con valores relativamente cercanos al valor real. Las correlaciones restantes muestran valores que divergen bastante de la media real, por encima de lo registrado por los modelos de ML. Para el valor mínimo reportado, el modelo de *SVR* tiene el valor más cercano al real, sin embargo, tanto las correlaciones, como los modelos de ML tienen datos similares. Finalmente, para el valor máximo, el modelo de *SVR*, es aquel que tiene una menor diferencia respecto al valor verdadero, aunque la diferencia respecto a los demás modelos es similar. Lo mismo se cumple para todas las correlaciones excepto para la desarrollada por Dokla y Osman, 1992, aunque la diferencia con respecto al mejor modelo de ML sigue siendo amplia.

Tabla 5.8: Descripción estadística de B_{ob} real y generado

Modelo		μ, [rb/STB]	d.e., [rb/STB]	min, [rb/STB]	max, [rb/STB]
Real		1.3354	0.3885	1.0000	3.2076
ML	Extra Trees	1.3226	0.3726	1.0337	2.9231
	SVR	1.3422	0.3759	0.9800	3.0361
	ANN	1.3397	0.3438	1.0509	2.8781
	SVR $p_b \rightarrow B_o$	1.3275	0.3759	1.0281	2.9456
	Extra Trees $p_b \rightarrow B_o$	1.3358	0.3437	1.0523	2.8757
	Random Forest $p_b \rightarrow B_o$	1.3305	0.3271	1.0523	2.7083
Correlaciones	Standing	1.3476	0.3938	1.0340	3.2076
	Glasø	1.3064	0.3541	1.0233	2.8431
	Al-Marhoun	1.3292	0.3260	1.0173	2.8145
	Dokla & Osman	1.3720	0.4071	0.9760	3.3041
	Petrosky & Farshad	1.3417	0.3744	1.0301	3.0003

Ahora, para el segundo bloque de algoritmos de ML donde se incluye la p_b generada, se tiene la menor diferencia en la media respecto al valor verdadero para el algoritmo de *Extra Trees*, esta diferencia es del orden de magnitud de 10^{-4} rb/STB, para los demás algoritmos y correlaciones esta diferencia aumenta al menos en un orden de magnitud. En cuanto al valor mínimo registrado se tiene que prácticamente todos los modelos comparten datos similares y, las correlaciones presentan valores más cercanos al valor real. Finalmente, para el valor máximo se puede notar que los algoritmos muestran diferencias más altas respecto al valor real al ser comparadas con las correlaciones.

5.3.3. Error relativo, error absoluto y coeficiente de correlación

En esta sección se muestran los errores relativos y absolutos, así como el coeficiente de correlación (R^2) registrado para los modelos que predicen B_{ob} y aquellos que toman a p_b como dato para predecir a B_{ob} (véase tabla 5.9).

Tabla 5.9: Errores relativos, absolutos y R^2 para los modelos de B_{ob}

	Modelo	Error relativo, [%]				Error absoluto, [rb/STB]				R^2, [%]
		μ	d.e.	min	max	μ (MAE)	d.e.	min	max	
ML	Extra Trees	4.27	4.17	0.00	25.07	0.0567	0.0613	0.0001	0.2845	95.35
	SVR	3.66	3.87	0.04	20.18	0.0487	0.0504	0.0007	0.2096	96.72
	ANN	3.09	3.81	0.01	16.65	0.0438	0.0645	0.0001	0.3394	95.95
	SVR $p_b \rightarrow B_o$	2.87	3.48	0.01	15.67	0.0388	0.0542	0.0001	0.2752	97.04
	Extra Trees $p_b \rightarrow B_o$	2.91	3.74	0.00	16.76	0.0415	0.0640	0.0000	0.3399	96.12
	Random Forest $p_b \rightarrow B_o$	3.27	4.46	0.05	19.83	0.0487	0.0897	0.0006	0.4993	93.07
Correlaciones	Standing	1.94	3.86	0.00	17.28	0.0239	0.0516	0.0000	0.3159	97.85
	Glasø	3.40	2.89	0.00	13.97	0.0481	0.0541	0.0000	0.3645	96.50
	Al-Marhoun	3.30	3.80	0.05	18.10	0.0468	0.0626	0.0006	0.3931	95.92
	Dokla & Osman	4.43	4.24	0.03	20.10	0.0563	0.0517	0.0003	0.2863	96.10
	Petrosky & Farshad	2.50	3.12	0.05	15.53	0.0328	0.0417	0.0010	0.2073	98.12

Se comienza analizando el primer bloque de modelos, en este caso, la media de error relativo va de 3.09 % para *ANN* a 4.27 % para *Extra Trees*. Las correlaciones registran valores que van del 1.94 % para M. B. Standing, 1977 hasta 4.43 % para Dokla y Osman, 1992, es decir, en este rubro, las correlaciones mantienen valores menores respecto a los algoritmos y, la propuesta por M. B. Standing, 1977, supera a todos los casos. Es importante mencionar que esta correlación se usó en el preprocesamiento de datos en el capítulo pasado. El valor relativo mínimo se mantiene consistente en la mayoría de modelos de ML y en las correlaciones. Respecto al error relativo máximo, el mejor modelo es el de *ANN* y, en ningún caso el error relativo máximo supera el 26 %, las correlaciones se colocan por debajo de este valor, ya que reportan errores no mayores al 21 %.

En cuanto a errores absolutos, la correlación de M. B. Standing, 1977 es la que presenta el valor más reducido, sin embargo, los modelos de ML arrojan valores del mismo orden de magnitud. En cuanto a la desviación estándar prácticamente todos los modelos y las correlaciones reportan datos similares. Cabe destacar que el error máximo más pequeño lo tiene el algoritmo de *SVR*, superando a las correlaciones, excepto la desarrollada por Petrosky y Farshad, 1993 que registra valores similares.

Finalmente, para el coeficiente de correlación (R^2) se tienen valores por encima del 95 % para los tres modelos de ML. El algoritmo de *SVR* tiene el valor más alto con un 96.72 %. Las correlaciones reportan valores significativamente altos y, solamente la correlación de Petrosky y Farshad, 1993 es capaz de superar la mejor estimación de los modelos de ML (*SVR*) por una diferencia de 1.4 %.

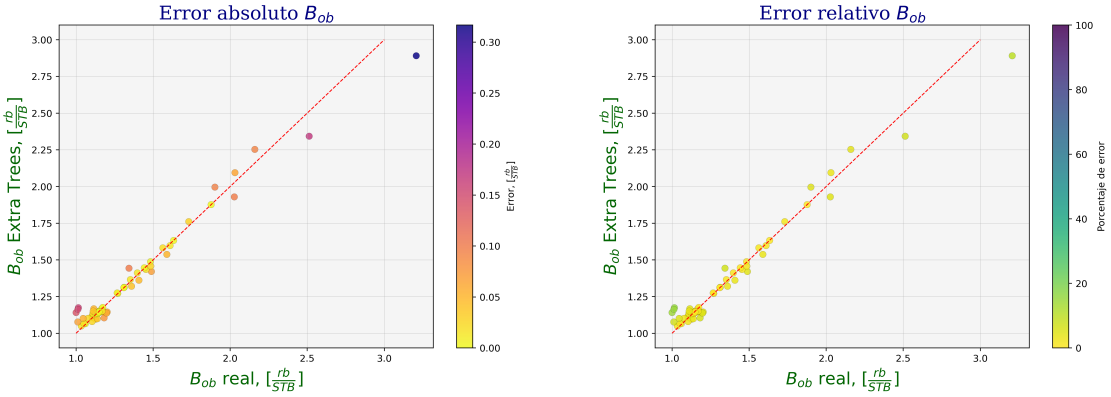


Figura 5.13: Error del modelo *Extra Trees* para B_{ob}

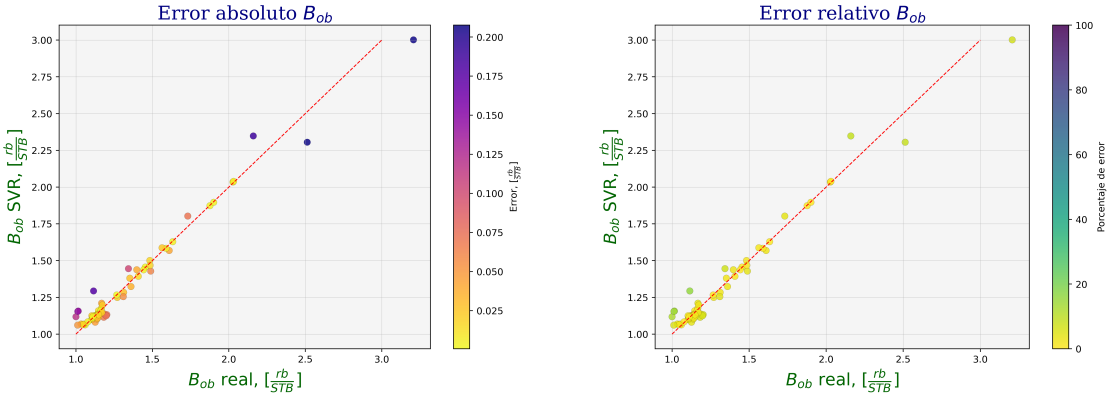


Figura 5.14: Error del modelo *SVR* para B_{ob}

Para el segundo bloque de algoritmos, se tienen resultados similares, el algoritmo de *SVR* es aquel con el menor error relativo (2.87 %), sin embargo, como se mencionó con anterioridad, es la correlación de M. B. Standing, 1977, la que reporta el menor error relativo medio. Los valores de error relativo mínimo son muy reducidos y, relativamente similares para los modelos de ML y las correlaciones. Para el error relativo máximo, la correlación de Glasø, 1980, supera a los modelos de ML, las demás correlaciones se mantienen a la par de los modelos.

En cuanto al error absoluto, las correlaciones se mantienen dentro del mismo rango de valores que los modelos de ML. El comportamiento es similar para el error absoluto mínimo. Al observar el valor máximo del error absoluto, las correlaciones muestran valores máximos de error menores a los reportados por los modelos de ML, especialmente la correlación escrita por Petrosky y Farshad, 1993.

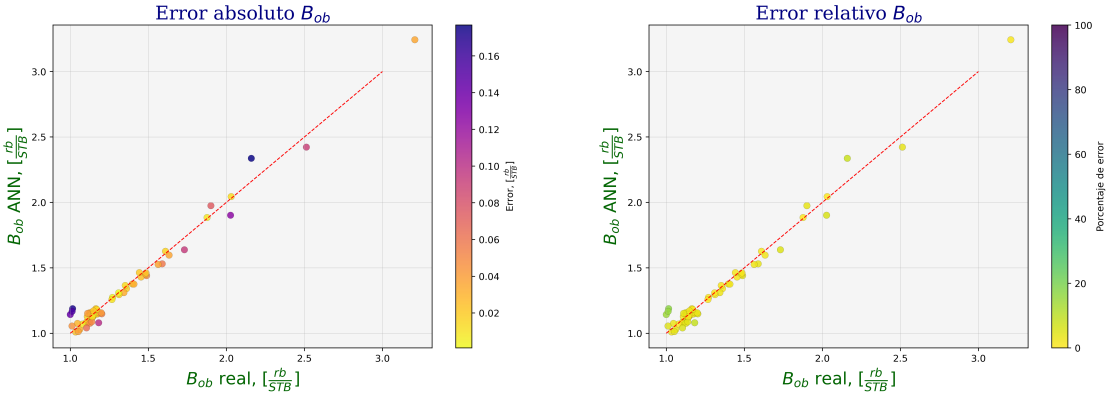


Figura 5.15: Error del modelo *ANN* para B_{ob}

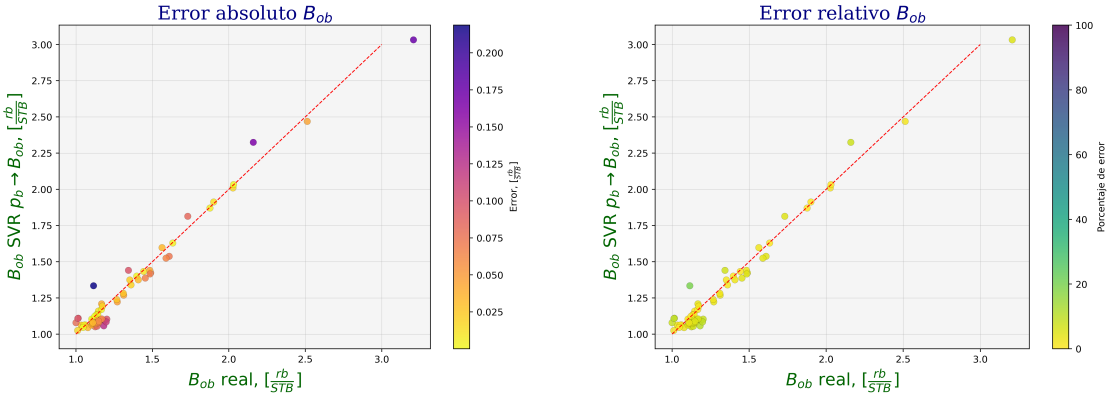


Figura 5.16: Error del modelo *SVR* $p_b \rightarrow B_{ob}$ para B_{ob}

Por último, se tienen los valores de R^2 para este bloque de algoritmos de ML. El algoritmo de *SVR* registra el valor más alto con un 97 %, este modelo supera a las correlaciones de Glasø, 1980, Al-Marhoun, 1988 y Dokla y Osman, 1992. Como ya fue establecido, la correlación de Petrosky y Farshad, 1993, se desempeña extraordinariamente bien en este rubro y, nuevamente no es superada por ningún algoritmo de ML, manteniendo una diferencia de alrededor del 1 % respecto al algoritmo de *SVR*. De cualquier forma, tanto el desempeño de los algoritmos de este bloque como los del anterior reportan valores bastante buenos, ya que, un coeficiente de correlación por encima del 95 %, se considera como un valor bastante aceptable.

A diferencia de la p_b, esta propiedad (B_{ob}) se encuentra descrita de una manera más precisa en las correlaciones presentadas, bajo todas las métricas éstas se mantienen a la par respecto a los algoritmos de ML en todos los casos aplicados. Cabe destacar que esta diferencia es muy reducida y su capacidad para estimar B_{ob}, es similar tanto para las correlaciones como para los modelos de ML.

En las figuras 5.13, 5.14, 5.15, 5.16, 5.17 y 5.18, se muestran gráficos donde se puede apreciar de manera cualitativa el error absoluto y relativo para los algoritmos de ML presentados en esta sección, además, a modo de comparación se presentan los mismos gráficos para las correlaciones de Petrosky y Farshad, 1993 (véase figura 5.19), y M. B. Standing, 1977 (véase figura 5.20), las dos correlaciones con el coeficiente de correlación más alto al estimar B_{ob}.

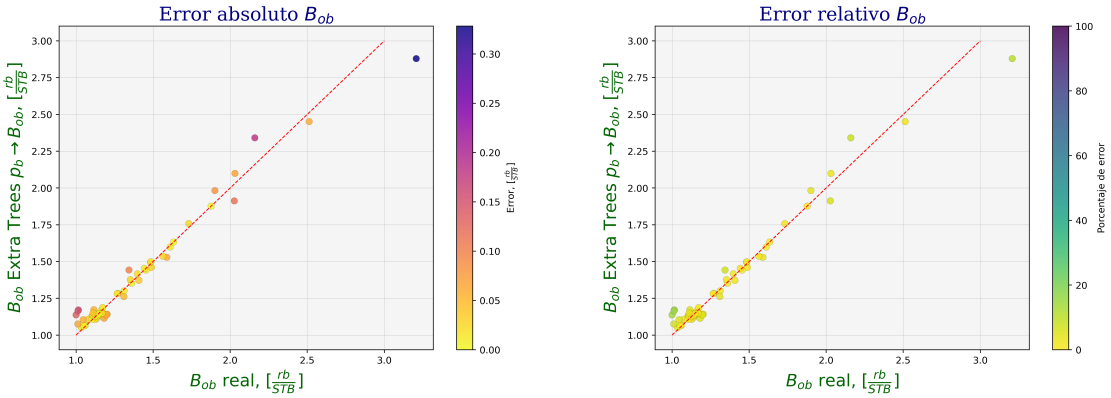


Figura 5.17: Error del modelo *Extra Trees* $p_b \rightarrow B_{ob}$ para B_{ob}

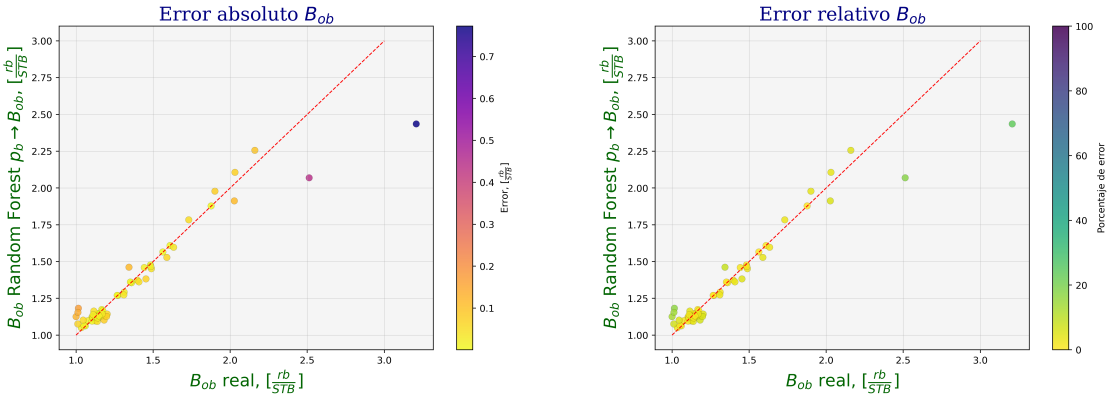


Figura 5.18: Error del modelo *Random Forest* $p_b \rightarrow B_{ob}$ para B_{ob}

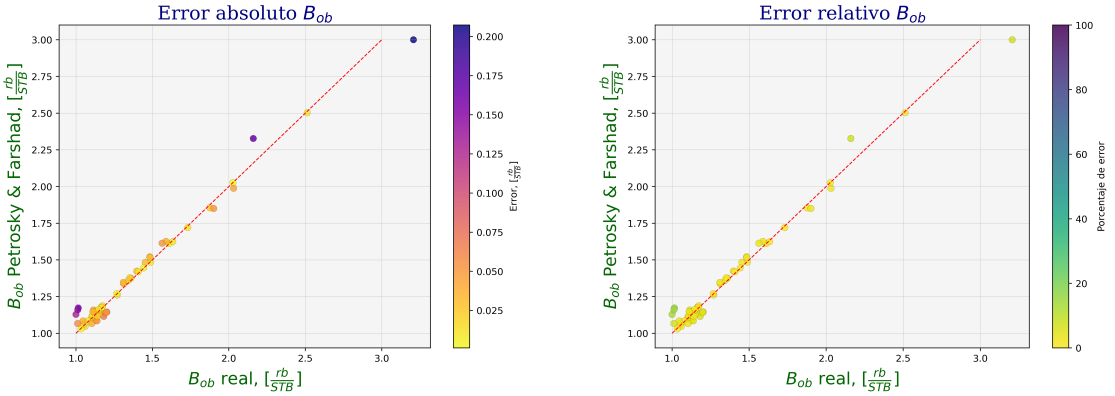


Figura 5.19: Error de la correlación de Petrosky y Farshad, 1993 para B_{ob}

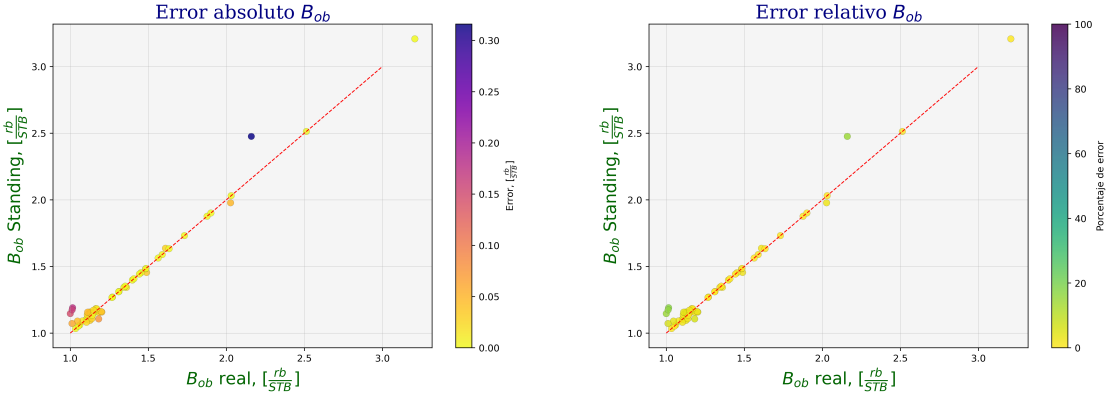


Figura 5.20: Error de la correlación de M. B. Standing, 1977 para B_{ob}

En las figuras 5.21, 5.22 y 5.23 se muestran el error relativo, el error medio absoluto (MAE) y el coeficiente de correlación (R^2), respectivamente para los modelos encargados de predecir B_{ob}. El error relativo (véase figura 5.21) de los algoritmos de *ANN*, *SVR* $p_b \rightarrow B_{ob}$ y *Extra Trees* $p_b \rightarrow B_{ob}$ es menor al valor medio de todos los modelos. La correlación propuesta por M. B. Standing, 1977y Petrosky y Farshad, 1993son las únicas dos que se mantienen con un error relativo menor a la media.

En la figura 5.22, se puede ver que para el caso de los modelos de *ANN*, *SVR* $p_b \rightarrow B_{ob}$ y *Extra Trees* $p_b \rightarrow B_{ob}$, se tiene un error medio absoluto menor a la media de todos los modelos. Además, el error medio de la correlación propuesta por M. B. Standing, 1977, es bastante reducido en comparación a todos los demás modelos y correlaciones, solamente la correlación de Petrosky y Farshad, 1993, mantiene valores igualmente por debajo de la media.

Para la figura 5.23, se aprecia que solamente dos algoritmos registran valores mayores a la media (esto se ve reflejado con un valor menor en la gráfica, ya que presenta a $1 - R^2$ en el eje de las ordenadas), estos modelos son: SVR y $SVR\ p_b \rightarrow B_{ob}$. En general, el desempeño de los modelos de ML es bastante bueno, aunque la correlación de Petrosky y Farshad, 1993, reporta resultados sobresalientes, siendo la que ostenta el valor más alto de R^2 en este trabajo. Cualitativamente se puede observar que los modelos de ML son prácticamente igual de competentes para predecir B_{ob} que las correlaciones disponibles en la literatura, incluso superando a algunas de estas.

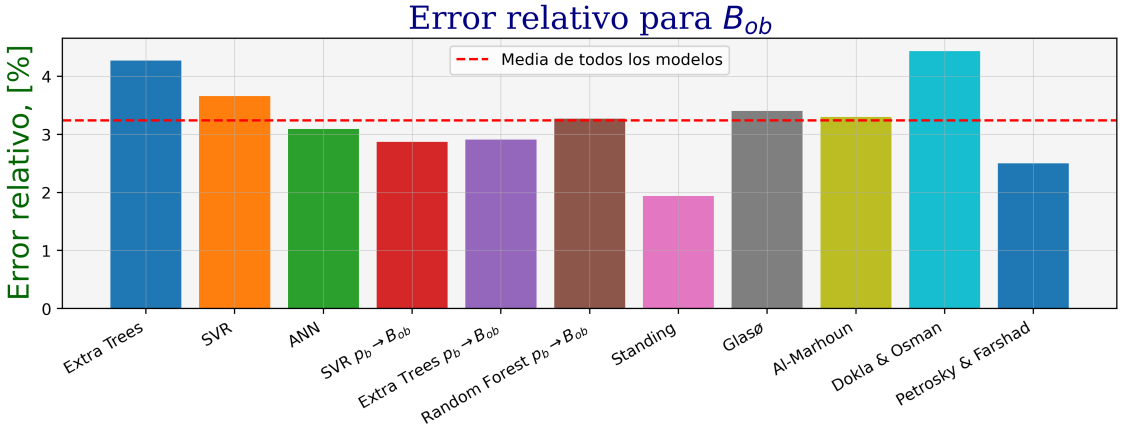


Figura 5.21: Error relativo (MAE) para B_{ob}

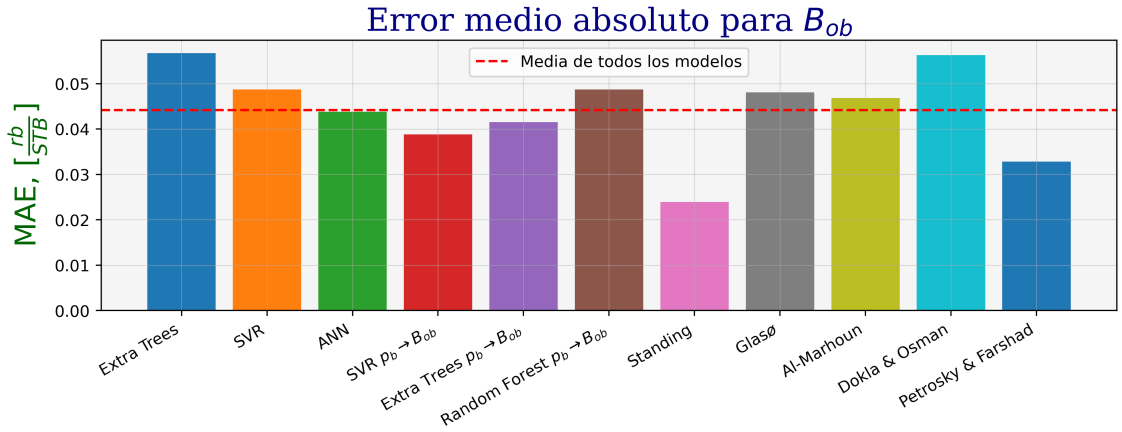


Figura 5.22: Error medio absoluto (MAE) para B_{ob}

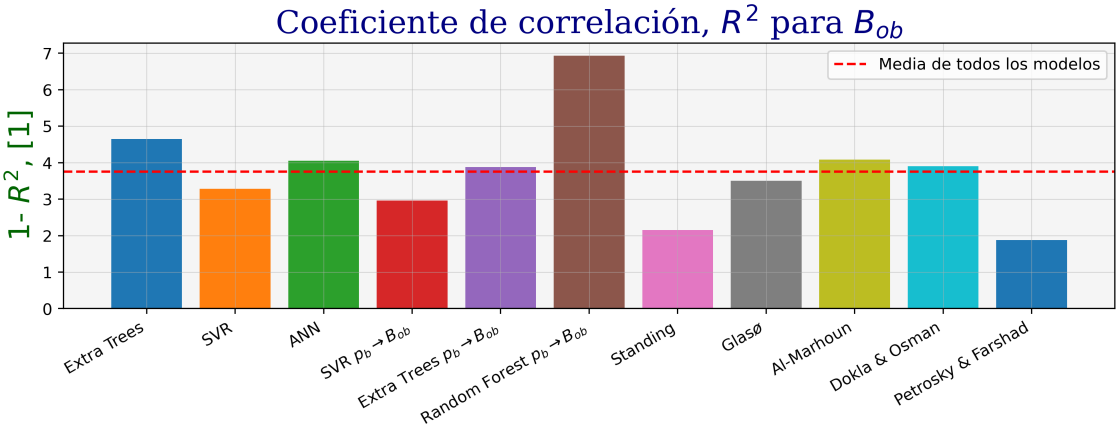


Figura 5.23: Coeficiente de correlación (R^2) para B_{ob}

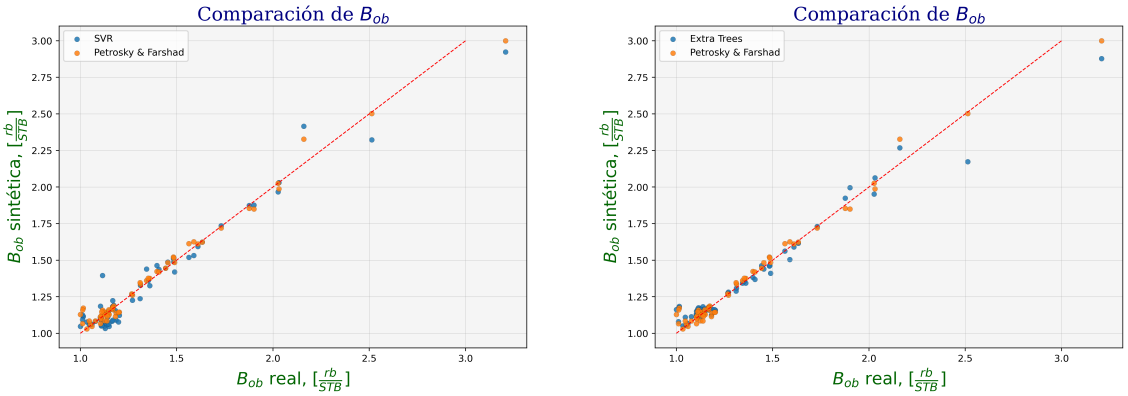
5.3.4. Comparación con la correlación de Petrosky y Farshad, 1993

Finalmente, en la figura 5.24 se muestra una comparación entre el mejor modelo para predecir B_{ob} (véase figura 5.24a), como el peor modelo de los tres seleccionados (véase figura 5.24b), ambos contra la mejor correlación de acuerdo a su valor de R^2. Para ambas figuras se puede ver que la dispersión de los puntos del modelo de ML es mayor a los puntos de la correlación, especialmente para valores mayores a 1.8 rb/STB, además de esto, los puntos por encima de 2.5 rb/STB se encuentran mucho más alejados para el algoritmo de *Extra Trees* en la figura 5.24b que para el algoritmo *SVR* en la figura 5.24a. En ambos casos el ajuste que muestra la correlación respecto a los datos reales es mayor.

De manera similar, se presenta la figura 5.25 donde se registra el comportamiento de los modelos de ML para predecir B_{ob}, tomando en cuenta la p_b estimada. La figura 5.25a, muestra el comportamiento para el modelo con el valor de R^2 más alto contra la correlación de Petrosky y Farshad, 1993, mientras que en la figura 5.25b se muestra el comportamiento del algoritmo con el valor de R^2 más bajo contra la misma correlación. Se puede ver que el ajuste del modelo *SVR*, es muy similar al de la correlación, ajustándose bien a los valores verdaderos de B_{ob}, por otro lado, el algoritmo de *Random Forest*, muestra puntos más dispersos desde los 1.3 hasta los 2.3 rb/STB, por encima de este valor, el algoritmo pierde precisión y muestra un mayor desajuste respecto a la correlación.

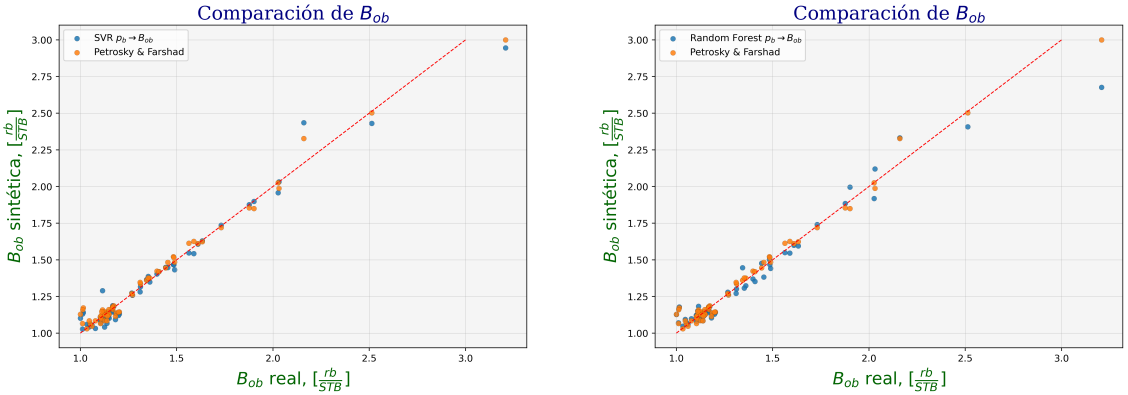
5.4. Análisis

Con base en los resultados obtenidos y en las secciones pasadas, se pueden extraer la siguientes observaciones respecto a los algoritmos de ML. Para la p_b, los tres algoritmos implementados registraron estimaciones más precisas que las correlaciones, en la mayoría de los casos lograron reducir el error relativo y el error absoluto medio. Además, el error absoluto máximo es mucho menor para los algoritmos que para las correlaciones, esto se traduce en una mayor confianza en



(a) *SVR* para B_{ob} (b) *Extra Trees* para B_{ob}

Figura 5.24: Comparación de los modelos de ML vs correlación de Petrosky y Farshad, 1993 para B_{ob}



(a) *SVR* $p_b \rightarrow B_{ob}$ para B_{ob} (b) *Random Forest* $p_b \rightarrow B_{ob}$ para B_{ob}

Figura 5.25: Comparación de los modelos de ML $p_b \rightarrow B_{ob}$ vs correlación de Petrosky y Farshad, 1993 para B_{ob}

las predicciones, ya que, incluso cuando la estimación no es precisa, el error esperado tiende a ser más conservador, con errores máximos de 1,900 psi. Por su parte, las correlaciones, por su parte, aumentan el error máximo esperado hasta 2,300 psi, llegando incluso a tener una diferencia de más de 3,000 psi respecto al valor verdadero. Por otro lado, el coeficiente de correlación mostró que, en todos los casos, lo modelos de ML superan a las correlaciones tradicionales, por 11.8 % (para el algoritmo de *Extra Trees*) en el mejor de los casos, aunado a esto, los modelos de ML mostraron un valor de R^2 por encima del 93 %, lo que es señal de una buena precisión para estimar p_b.

Es importante mencionar que, el algoritmo de *Extra Trees* fue el que mostró los resultados más consistentes bajo todas las métricas con un valor de R^2 de 95.2 %. En segundo lugar, el otro algoritmo de la familia de árboles de decisión (*Random Forest*) registró resultados ligeramente menores pero igual de competitivos, alcanzando valores de R^2 aproximadamente un 1 % menores a los del algoritmo anterior. En conjunto, estos dos algoritmos muestran una capacidad de estimar p_b bastante prometedora. El algoritmo de *KNN*, presentó estimaciones con valores cercanos al 90 %, por lo que se mantiene como buen candidato para predecir p_b.

Profundizando ahora sobre la segunda variable objetivo, los modelos de ambos bloques encargados de predecir B_{ob}, registraron errores relativos mayores a los presentados por las correlaciones, mostrando, en el mejor de los casos, una diferencia de casi 1 % respecto a la correlación con el mejor valor. En cuanto al error absoluto medio, dos de las cinco correlaciones superan a los modelos de ML de ambos bloques con una diferencia del orden de 10^{-2} rb/STB, lo que coloca a las correlaciones por encima de los modelos bajo esta métrica, sin embargo, este valor es muy reducido, por lo que es poco significativa la diferencia. Además de lo mencionado, el error absoluto máximo indica que, los algoritmos más competitivos de los dos bloques, pueden compararse a las correlaciones, registrando valores similares, por lo que, mantienen la misma fiabilidad para mantener diferencias máximas respecto al valor real del mismo orden de magnitud que las correlaciones empíricas. Finalmente, para el valor de R^2, dos correlaciones superaron a todos los algoritmos; sin embargo, esta diferencia es menor al 1 % en más de un caso, por lo que, se mantienen con una capacidad similar a estas correlaciones, además de que superaron a las otras tres. Por todo lo antes mencionado se puede notar que los modelos de ML se mantienen dentro del mismo rango de valores que las correlaciones, por lo que, pueden ser optimizados para alcanzar valores que superen a estas.

Analizando el desempeño de cada bloque de algoritmos, se encontró que, los modelos que no tomaron en cuenta la p_b estimada, presentan un error relativo promedio menor en aproximadamente 0.72 %, una diferencia del error absoluto de $9x10^{-3}$ rb/STB a favor de los modelos que no toman a p_b y, siguiendo esta tendencia, el valor de R^2 registrado es en promedio más alto para el primer bloque de algoritmos en un 1.65 %. Por esto, se puede notar que los algoritmos que no tomaron la p_b estimada como dato de entrada, tienden a una mayor precisión en sus estimaciones, las diferencias son menores, pero implica que no hay necesidad de predecir a p_b para estimar a B_{ob}, al menos mediante el ML.

Por último, prácticamente todos los algoritmos mostraron resultados consistentes, solamente el algoritmo de *Random Forest* $p_b \rightarrow B_{ob}$, tuvo menor precisión. Fuera de este caso, todos los algoritmos registran valores que pueden considerarse equivalentes entre sí, lo que abre la puerta a que estos algoritmos puedan optimizarse para obtener predicciones que superen a las correlaciones con alta precisión.

Capítulo 6

Conclusiones

En esta tesis se presentó una evaluación de técnicas de *Machine Learning* (ML) para la estimación de las propiedades PVT (p_b y B_{ob}). Con ayuda del lenguaje de programación Python y bibliotecas enfocadas en ML y manejo de datos, se aplicaron seis algoritmos de aprendizaje supervisado enfocados a problemas de regresión a tres escenarios distintos y, se realizó la evaluación cualitativa y cuantitativa de los resultados obtenidos con un set de datos de validación que incluye un conjunto de 26 datos de campos mexicanos y 39 datos pertenecientes a yacimientos de otras partes del mundo; los resultados se evaluaron contra cinco de las correlaciones más empleadas en la ingeniería petrolera.

Presión de burbuja

En conjunto, los tres algoritmos de ML (*Extra Trees*, *Random Forest* y *KNN*) evaluados para la predicción de la p_b, demostraron un desempeño por encima de las correlaciones empíricas bajo prácticamente todas las métricas empleadas. Los algoritmos de la familia de árboles de decisión (*Extra Trees* y *Random Forest*) lograron reducir el error relativo registrado por las correlaciones desde 25 % (correlación de Al-Marhoun, 1988) hasta un 22 % (*Random Forest*) y 21 % (*Extra Trees*), el error medio absoluto desde 428 (correlación de M. B. Standing, 1977) hasta 272 psi (*Random Forest*) y 236 psi (*Extra Trees*). Los tres algoritmos presentaron un valor de R^2 mayor respecto a la mejor estimación de una correlación (M. B. Standing, 1977 con $R^2 = 83.44$); *Extra Trees* con $R^2 = 95.24$, *Random Forest* con $R^2 = 93.93$ y *KNN* con $R^2 = 88.91$.

Factor de volumen de formación del aceite a la presión de burbuja

Los algoritmos que no tomaron a la presión de burbuja sintética para predecir a B_{ob}, mostraron un desempeño ligeramente superior respecto a los modelos que estimaron a B_{ob} tomando solamente los datos de entrada originales, sin embargo, el impacto de esto fue relativamente menor en las predicciones, el valor de R^2, aumentó en un 0.6 % en promedio al no tomar a p_b como dato de entrada; debido a esto, la presión de burbuja no afecta significativamente la estimación de B_{ob} mediante algoritmos de ML. Los algoritmos de *Extra Trees*, *SVR*, *ANN*, *SVR* $p_b \rightarrow B_{ob}$ y *Random Forest* $p_b \rightarrow B_{ob}$ registraron valores de R^2 mayores al 95 % . De acuerdo con la idea anterior, los modelos de ML lograron una capacidad para predecir B_{ob} que es similar a las estimaciones de las correlaciones con un valor de R^2 más elevado (M. B. Standing, 1977con $R^2 = 97.85$ y Petrosky y Farshad, 1993 con $R^2 = 98.12$) Esta tendencia se mantiene con el error relativo y el error medio, ya que los algoritmos

de ML presentan una diferencia promedio en el error relativo respecto a las correlaciones del 0.55 % y de 10^{-3} rb/STB para el error absoluto. Estas diferencias son despreciables, por lo que los algoritmos de ML se colocan a la par de las correlaciones empíricas para estimar B_{ob}.

La evaluación de los algoritmos de ML sobre el conjunto de datos de validación, de los cuales el 40 % de los datos corresponden a yacimientos mexicanos, arrojó que la precisión de los algoritmos es, en el peor de los casos equiparable al de una correlación empírica (véase apéndice D). En esta tesis el set de datos disponía de casi 500 valores para cada propiedad, esto tuvo como consecuencia, que los modelos de ML se entrenaran con alrededor de 300 valores (conjunto de entrenamiento). La precisión de un algoritmo está ligada al volumen de datos usado, por lo que, en principio, estos modelos deberían tener la capacidad de aumentar significativamente la precisión de sus estimaciones a medida que la cantidad de datos empleada para entrenar al algoritmo aumente.

Con base en lo anterior, se concluye que el ML tiene capacidades de aplicación en la predicción de propiedades PVT para yacimientos mexicanos y campos de otras regiones del mundo. Se propone su implementación mediante el desarrollo de una herramienta computacional sencilla y portable, tomando como base, el código mostrado en el apéndice B de este trabajo.

6.1. Aplicación del *Machine Learning* en la predicción de propiedades PVT

Derivado de los resultados obtenidos en este trabajo, se propone la implementación del ML a la predicción de propiedades PVT con base en los siguientes puntos:

1. **Los resultados obtenidos con ML son iguales o superiores a las correlaciones empíricas**, principalmente para la estimación de la p_b, donde superan a los métodos tradicionales.

2. **Su rango de aplicación no se encuentra limitado a una sola región**, ya que en este trabajo, los resultados obtenidos corresponden a datos de yacimientos mexicanos así como a campos de otras partes del mundo.

3. **La precisión de sus resultados tiende a mejorar**, a medida que la base de datos que el algoritmo tiene a su disposición crece, la capacidad de predicción aumenta. En este trabajo se emplearon menos de 500 datos y los resultados están al menos a la par de los métodos utilizados actualmente.

4. **Los resultados de varios algoritmos pueden apilarse para una mejor estimación**, al usar varios algoritmos con resultados similares, sus predicciones pueden apilarse (es decir, promediar la estimación de cada modelo con las predicciones de los otros modelos) para disminuir el error de cada algoritmo.

5. **El ML es de fácil implementación**, a pesar de requerir ciertos conocimientos esenciales, su implementación es sencilla, ya que existen módulos (en Python) con una diversidad de algoritmos y herramientas relacionadas con el ML listas para aplicarse.

Por estas razones, el ML se presenta como una alternativa atractiva frente a los métodos tradicionales para determinar propiedades PVT. Además, al ser calibrado adecuadamente, un modelo de ML puede ofrecer una precisión mayor a las correlaciones empíricas.

6.2. Observaciones y recomendaciones

De los seis algoritmos empleados en este trabajo, el de *Extra Trees* mostró un desempeño más consistente y una capacidad para estimar a las dos variables objetivo (p_b y B_{ob}) en los escenarios planteados. En general, la familia de algoritmos de árboles de decisión se acoplaron muy bien para la estimación de la p_b, con resultados que superaron a las correlaciones tradicionales. Para ambos algoritmos de esta familia (*Extra Trees* y *Random Forest*), se recomienda fijar como parámetro interno el *error absoluto*. Además, se observaron estimaciones precisas a partir de los 100 árboles de decisión, por esto, se recomienda no aumentar la cantidad de árboles de decisión ya que este parámetro afecta considerablemente el tiempo de entrenamiento del algoritmo a cambio de un ligero aumento en la precisión de este. Finalmente, el parámetro de profundidad máxima mostró resultados ampliamente variables, por lo que se sugiere una evaluación cuidadosa de su impacto en las estimaciones.

Respecto a B_{ob}, el algoritmo más consistente fue el de *SVR*, sin embargo, tuvo capacidad insuficiente para adaptarse a la predicción de p_b, mostrando resultados muy por debajo de la media. Es recomendable variar independientemente los parámetros internos de C y nu para evaluar su impacto sobre las estimaciones. De manera general, este algoritmo no exige muchos recursos de procesamiento, por lo que una búsqueda de malla extensa no representa un costo computacional significativo.

El modelo de *ANN* arrojó estimaciones prometedoras para B_{ob}, además al ser un algoritmo altamente flexible, principalmente en su arquitectura. Se recomienda una investigación más amplia en torno a su capacidad para la predicción de propiedades PVT. Al tener tantas combinaciones posibles de hiperparámetros, se recomienda realizar una búsqueda de malla segmentada para cada uno de estos hiperparámetros. Aprovechando, que el costo computacional de entrenamiento para este algoritmo no es muy alto (de manera similar al modelo de *SVR*).

De la familia de algoritmos de *Nearest Neighbors* se recomienda solamente el uso de *KNN*, ya que su otra variante (*Radius Neighbors*), carece de las capacidades para predecir propiedades PVT (al menos con cantidad de datos recopilados para este trabajo). Para el algoritmo de *KNN*, se recomienda emplear como parámetro interno de solución el algoritmo *ball tree*, ya que al ser aplicado, mostró resultados consistentes para p_b y B_{ob}. Por otro lado, el parámetro de número de vecinos debe ser calibrado con base a la densidad de datos disponible.

En cuanto a las transformaciones de datos, en este trabajo solo se mencionaron dos tipos, de escalamiento y PCA; sin embargo, existe una gran diversidad de transformaciones que pueden ser de utilidad con base en el tipo de datos y algoritmos disponibles. Una buena práctica consiste en tomar distintas transformaciones y estudiar su efecto sobre las estimaciones. Para la predicción de la p_b y B_{ob}, se recomienda evaluar la opción de escalamiento mediante estandarización o máximo-absoluto, ya que fueron las dos transformaciones de escalamiento con mayor presencia en los resultados. Por otra parte, la transformación PCA no mostró tener un impacto positivo en las estimaciones de las propiedades PVT, salvo para el algoritmo de *ANN*.

Apéndice A

Correlaciones

En este apéndice, se presentan las correlaciones empleadas en este trabajo para comparar los datos obtenidos mediante los diferentes algoritmos de ML para la predicción de p_b y B_{ob}.

Las unidades empleadas en las correlaciones son las siguientes:

$$\text{Densidad API } (^\circ API) : 1$$

$$\text{Densidad relativa del gas } (\gamma_g) : 1$$

$$\text{Densidad relativa del aceite } (\gamma_o) : 1$$

$$\text{Temperatura } (T) :^\circ F$$

$$\text{Relación de solubilidad } (R_s) : \frac{SCF}{STB}$$

$$\text{Presión de burbuja } (p_b) : psia$$

$$\text{Factor de volumen de formación a la presión de burbuja } (B_{ob}) : \frac{rb}{STB}$$

El sufijo b en una propiedad indica que esta se encuentra en la presión de burbuja.

A.1. M. B. Standing, 1977

Para el desarrollo de esta correlación el autor usó un set de 105 datos determinados experimentalmente pertenecientes a 22 mezclas de fluidos provenientes a campos de California en los Estados Unidos. En la tabla A.1 se muestra el rango de valores de este set de datos.

Tabla A.1: Rango de valores usados para el desarrollo de la correlación de M. B. Standing, 1977, modificado de Bánzer, 1996, p.50

Parámetro	Unidad	Valor
Densidad API	°API	16.5 - 69.8
Temperatura	°F	100 - 258
Densidad relativa del gas	1	0.59 - 0.95
Relación de solubilidad	SCF/STB	20 - 1425
Presión de burbuja	psi	130 - 7000
Factor de volumen del aceite	rb/STB	1.024 - 2.15

p_b

$$F = \frac{R_{sb}}{\gamma_g}^{0.83} 10^{0.00071T - 0.0125^\circ API} \tag{A.1}$$

$$p_b = 18.2\,[F - 1.4] \tag{A.2}$$

B_{ob}

$$F = R_{sb}\sqrt{\frac{\gamma_g}{\gamma_o}} + 1.25T \tag{A.3}$$

$$B_{ob} = 0.9759 + 12 \times 10^{-5} F^{1.2} \tag{A.4}$$

A.2. Glasø, 1980

El autor empleó 45 muestras pertenecientes al mar del Norte en su mayoría, aunque también se incluyen datos de campos localizados en Estados Unidos y Argelia para el desarrollo de las correlaciones. En la tabla A.2 se muestran los rangos de los datos empleados por el autor.

Tabla A.2: Rango de valores usados para el desarrollo de la correlación de Glasø, 1980, modificado de Bánzer, 1996, p.55

Parámetro	Unidad	Valor
Densidad API	°API	22.3 - 48.1
Temperatura	°F	80 - 280
Densidad relativa del gas	1	0.65 - 1.276
Relación de solubilidad	SCF/STB	90 - 2637
Presión de burbuja	psi	165 - 7142
Factor de volumen del aceite	rb/STB	1.025 - 2.588

p_b

$$F = \frac{R_{sb}}{\gamma_g}^{0.816} \frac{T^{0.172}}{{}^{\circ}API^{0.989}} \tag{A.5}$$

$$p_b = 10^{\left[1.7669+1.7447\log_{10}F-0.30218(\log_{10}F)^2\right]} \tag{A.6}$$

B_{ob}

$$F = R_{sb}\left(\frac{\gamma_g}{\gamma_o}^{0.526}\right) + 0.968T \tag{A.7}$$

$$B_{ob} = 1 + 10^{\left[-6.58511+2.91329log_{10}F-0.27683(\log_{10}F)^2\right]} \tag{A.8}$$

A.3. Al-Marhoun, 1988

El autor presenta un set de 160 datos pertenecientes a 69 pozos del Medio Oriente. La descripción de este set de datos se puede observar en la tabla A.3.

Tabla A.3: Rango de valores usados para el desarrollo de la correlación de Al-Marhoun, 1988, modificado de Bánzer, 1996, p.58

Parámetro	Unidad	Valor
Densidad API	°API	19.4 - 44.6
Temperatura	°F	74 - 240
Densidad relativa del gas	1	0.752 - 1.367
Relación de solubilidad	SCF/STB	26 - 1602
Presión de burbuja	psi	20 - 3573
Factor de volumen del aceite	rb/STB	1.032 - 1.997

p_b

$$p_b = 5.38088 \times 10^{-3} R_{sb}^{0.715082} \ \gamma_g^{-1.87784} \ \gamma_o^{3.1437} \ (T+460)^{1.32657} \tag{A.9}$$

B_{ob}

$$F = R_{sb}^{0.74239} \ \gamma_g^{0.323294} \ \gamma_o^{-1.20204} \tag{A.10}$$

$$B_{ob} = 0.497069 + 0.862963 \times 10^{-3}(T+460) + 0.182594 \times 10^{-2}F + 0.318099 \times 10^{-5}F^{-2} \tag{A.11}$$

A.4. Dokla y Osman, 1992

Se recopilaron 51 muestras de fondo de pozo de mezclas de aceite de los Emiratos Árabes Unidos para las correlaciones presentadas a continuación. El rango del set de datos se muestra en la tabla A.4.

Tabla A.4: Rango de valores usados para el desarrollo de la correlación de Dokla y Osman, 1992, modificado de Bánzer, 1996, p.58

Parámetro	Unidad	Valor
Densidad API	°API	28.3 - 40.3
Temperatura	°F	190 - 275
Densidad relativa del gas	1	0.879 - 1.290
Relación de solubilidad	SCF/STB	81 - 2266
Presión de burbuja	psi	590 - 4640
Factor de volumen del aceite	rb/STB	1.216 - 2.493

p_b

$$p_b = 0.836386 \times 10^4 R_{sb}^{0.724047} \ \gamma_g^{-1.01049} \ \gamma_o^{0.107991} \ (T+460)^{-0.952584} \tag{A.12}$$

B_{ob}

$$F = R_{sb}^{0.773572} \ \gamma_g^{0.40402} \ \gamma_o^{-0.882605} \tag{A.13}$$

$$B_{ob} = 0.431935 \times 10^{-1} + 0.156667 \times 10^{-2}(T+460) + 0.139775 \times 10^{-2} F + 0.380525 \times 10^{-5} F^{-2} \tag{A.14}$$

A.5. Petrosky y Farshad, 1993

Las muestras empleadas para el desarrollo de las siguientes correlaciones pertenecen a yacimientos costa afuera ubicados en Texas y Louisiana, el set comprende 81 datos PVT. De manera similar a las demás correlaciones, se muestra el rango de valores del set de datos empleado para el desarrollo de las correlaciones en la tabla A.5.

p_b

$$F = \frac{R_{sb}^{0.5774}}{\gamma_g^{0.8439}} 10^{\left[4.561 \times 10^{-5} T^{1.3911} - 7.916 \times 10^{-4} \ {}^{\circ}API^{1.541}\right]} \tag{A.15}$$

$$p_b = 112.727 \left[F - 12.34\right] \tag{A.16}$$

Tabla A.5: Rango de valores usados para el desarrollo de la correlación de Petrosky y Farshad, 1993, modificado de Bánzer, 1996, p.59

Parámetro	Unidad	Valor
Densidad API	°API	16.3 - 45.0
Temperatura	°F	114 - 288
Densidad relativa del gas	1	0.5781 - 0.8519
Relación de solubilidad	SCF/STB	217 - 1406
Presión de burbuja	psi	1700 - 10692
Factor de volumen del aceite	rb/STB	1.1178 - 1.6229

B_{ob}

$$F = R_{sb}^{0.3738}\left(\frac{\gamma_g^{0.2914}}{\gamma_o^{0.6265}}\right) + 0.24626T^{0.5371} \tag{A.17}$$

$$B_{ob} = 1.0113 + 7.2046 \times 10^{-5} F^{3.0936} \tag{A.18}$$

Apéndice B

Códigos

En este apéndice se presenta, por medio de pseudocódigo, el desarrollo de la herramienta computacional generada para aplicar los algoritmos de ML en la predicción de p_b y B_{ob}.

En las siguientes líneas de código se presenta la función creada para obtener las variables de entrada de acuerdo a la variable objetivo y el set de datos de entrada, la variable x representa las variables de entrada, mientras que y representa a la variable objetivo.

```
import pandas
import numpy
import matplotlib
import scikit-learn

def get_variables(var):

    if var = 'pb':
        x = ('API', 'gamma_g', 'T', 'Rs')
        y = 'pb'

    if var = 'bo':
        x = ('API', 'gamma_g', 'T', 'Rs')
        y = 'bo'

    return x, y
```

```python
def Parameters(regressor):

    if regressor == 'Random Forest':
        parameters['n_estimators'] = [10, 50, 100, 400, 500, 1000]
        parameters['criterion'] = ['absolute_error',
                                   'squared_error', 'poisson']
        parameters['max_depth'] = [5, 15, 25, 30, 40, 45, 50, 100]
        parameters['min_samples_split'] = [0.5, 1.0, 1.5, 2.0, 5.0, 10.0]
        parameters['min_samples_leaf'] = [0.5, 1.0, 1.5, 2.0]

    if regressor == 'Extra Trees':
        parameters['n_estimators'] = [10, 50, 100, 400, 500, 1000]
        parameters['criterion'] = ['absolute_error',
                                   'squared_error', 'poisson']
        parameters['max_depth'] = [5, 15, 25, 30, 40, 45, 50, 100]
        parameters['min_samples_split'] = [0.5, 1.0, 1.5, 2.0, 5.0, 10.0]
        parameters['min_samples_leaf'] = [0.5, 1.0, 1.5, 2.0]

    if regressor == 'SVR':
        parameters['nu'] = [0.1, 0.25, 0.5, 0.75, 1.0]
        parameters['kernel'] = ['poly', 'rbf', 'sigmoid', 'precomputed']
        parameters['gamma'] = ['scale', 'auto']
        parameters['C'] = [0.5, 1.0, 1.5, 3.5, 4.0, 5.0]
        parameters['max_iter'] = [1000, 10000, 100000]
        parameters['tol'] = [1e-2, 1e-3, 1e-4]

    if regressor == 'ANN':
        parameters['activation'] = ['identity', 'logistic', 'tanh', 'relu']
        parameters['hidden_layer_sizes'] = [(2, 2), (2,), (4, 4),
                                            (4,), (6, 6), (6, )]
        parameters['max_iter'] = [1000, 4000, 10000]
        parameters['solver'] = ['lbfgs', 'sgd', 'adam']
        parameters['learning_rate_init'] = [1000, 4000, 10000]
        parameters['tol'] = [1e-2, 1e-3, 1e-4]

    if regressor == 'KNN':
        parameters['n_neighbors'] = [2, 4, 3, 8, 10]
        parameters['algorithm'] = ['ball_tree', 'auto', 'kd_tree', 'brute']

    if regressor == 'Radius Neighbors':
        parameters['radius'] = [0.5, 0.8, 1.0, 1.2, 2.0]
        parameters['weights'] = ['uniform', 'distance']
        parameters['algorithm'] = ['ball_tree', 'auto', 'kd_tree', 'brute']

# Transformaciones
parameters['scaler'] = ['passthrough', StandardScaler,
                        MinMaxScaler, Normalizer, MaxAbsScaler]
parameters['pca'] = ['passthrough', PCA(1), PCA(2), PCA()]

return parameters
```

Las líneas del código anterior generan un diccionario en Python que contiene la lista de híper parámetros sobre los cuales realiza una búsqueda de malla de acuerdo al algoritmo seleccionado, además de que incluye las dos transformaciones de los datos de entrada. Los valores para los hiperparámetros solamente son usados para ejemplificar los posibles valores que puede tomar cada uno. Se recomienda consultar la página de la biblioteca *Scikit-Learn* para mayor información respecto a los distintos parámetros de cada modelo, su impacto y los valores que puede tomar.

```
regressors = [Extra Trees, Random Forest, SVR, ANN, KNN, Radius Neighbors]
variables = ['pb', 'bo']

for regressor in regressors:
    parameters_regressor = Parameters(regressor)
    grid_search = cross_validation(regressor, parameters_regressor, k=5)

    for var in variables:
        x, y = get_variables(var)
        x_train, x_test, y_train, y_test = split(x, y)
        grid_search_fitted = grid_search.fit(x_train, y_train)
        best_regressor = grid_search_fitted.best_regressor(x_test, y_test)
        score_test = best_regressor.score(x_test, y_test)
        best_params = best_regressor.parameters

        x_vay, y_val = get_validation(var)
        predicted_values = best_regressor.predict(x)
        score_val = best_regressor.score(x_vay, y_val)
        predicted_values.to_excel
        best_params.to_excel
```

Finalmente, se muestran las líneas empleadas para realizar una búsqueda de malla iterando sobre los distintos regresores, es decir, los algoritmos de ML tomados para este trabajo, cada uno de estos proviene de la biblioteca de *Scikit-Learn*. Se tiene un segundo ciclo **for** anidado para los distintos escenarios, de acuerdo con las variables de entrada y la variable objetivo. Posteriormente se dividen estos valores en un set de entrenamiento y uno de validación, seguido de esto se realiza una búsqueda de malla y una validación *k-fold* con $k = 5$. De acuerdo con los resultados derivados de la búsqueda de malla, se selecciona la mejor combinación de híper parámetros y transformaciones, con lo cual se entrena el modelo y se generan las predicciones de la variable objetivo sobre el set de datos de validación; los *scores* de prueba y validación son obtenidos, respectivamente. Finalmente, estos valores se guardan en una hoja de cálculo en formato $.xlsx$, análogamente, la lista con los parámetros arrojados es guardada.

Se recomienda evaluar detenidamente el efecto de cada híper parámetro sobre las predicciones y el *score* obtenido sobre el set de datos de prueba para determinar que rango de valores es el más óptimo en la búsqueda de malla. Para una mayor apreciación de lo desarrollado en este apéndice, el proceso descrito se esquematiza en la figura B.1.

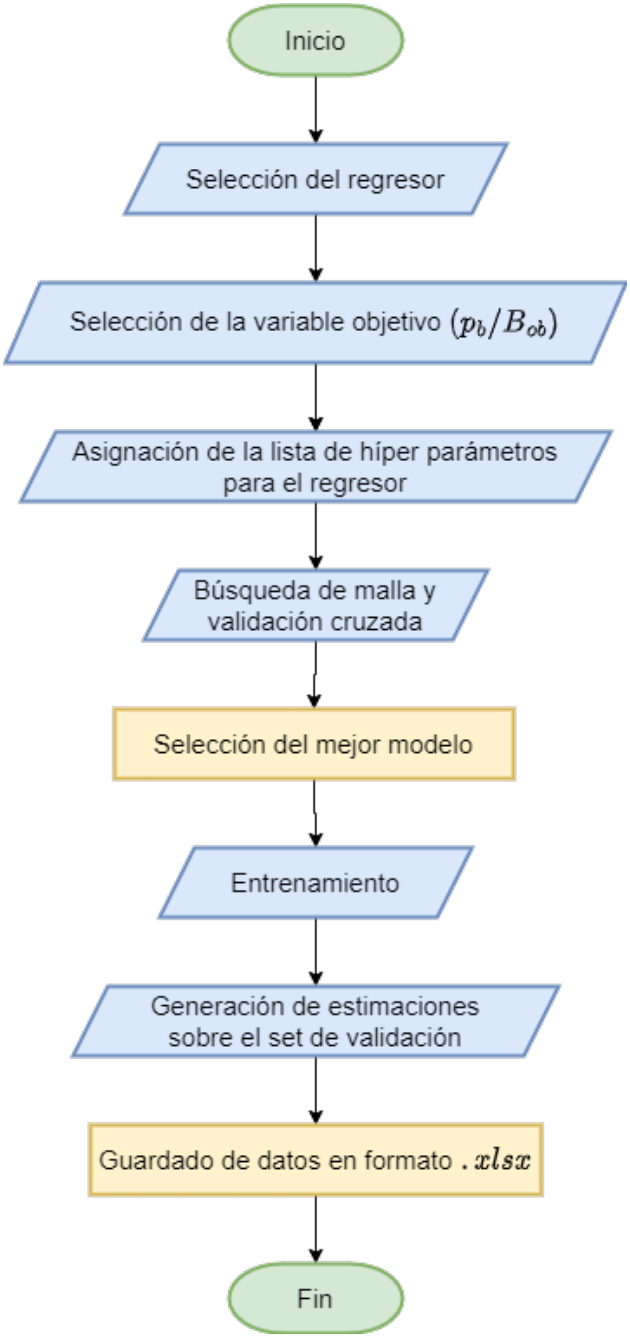


Figura B.1: Diagrama de flujo de los pasos seguidos en la herramienta computacional

Apéndice C

Datos generados

En aras de completitud para esta tesis y, para el lector interesado en profundizar más en la relación del ML con la predicción de propiedades PVT, se presenta la descripción estadística completa de los datos generados mediante los seis algoritmos de ML propuestos en este trabajo, de manera similar se muestran las tablas de error relativo, absoluto y coeficiente de correlación; estas tablas se muestran para p_b y B_{ob}; se añaden los resultados registrados por las correlaciones ya mencionadas.

Tabla C.1: Coeficiente de correlación (R^2) para el set de validación

Modelos	Coeficiente de correlación, %		
	p_b	B_{ob}	$p_b \rightarrow B_{ob}$
Extra Trees	95.24	95.35	96.12
Random Forest	93.93	97.45	93.07
SVR	8.08	96.72	97.04
ANN	67.62	95.95	82.33
KNN	88.91	92.89	84.92
Radius Neighbors	1.20	20.25	19.91

Tabla C.2: Descripción estadística completa de p_b real y generada

Modelo		μ, [psi]	d.e., [psi]	min, [psi]	max, [psi]
Real		1861	1613	107	6614
ML	Extra Trees	1899	1615	196	6274
	Random Forest	1858	1594	185	6299
	KNN	1991	1447	410	5746
	ANN	2039	91	1905	2265
	SVR	2208	122	2059	3162
	Radius Neighbors	2233	1541	-1258	8993
Correlaciones	Standing	1923	1841	24	9543
	Glasø	2406	1620	11	7877
	Al-Marhoun	1649	1876	39	9390
	Dokla & Osman	1292	1369	59	6745
	Petrosky & Farshad	1571	2107	-1159	9175

Tabla C.3: Descripción estadística completa de B_{ob} real y generado

Modelo		μ, [rb/STB]	d.e., [rb/STB]	min, [rb/STB]	max, [rb/STB]
Real		1.3354	0.3885	1.0000	3.2076
ML	Extra Trees	1.3226	0.3726	1.0337	2.9231
	SVR	1.3422	0.3759	0.9800	3.0361
	ANN	1.3397	0.3438	1.0509	2.8781
	Random Forest	1.3231	0.3164	1.0515	2.6741
	KNN	1.3404	0.3407	1.0769	2.6223
	Radius neighbors	1.4093	0.0674	1.3612	1.9320
	SVR $p_b \rightarrow B_o$	1.3275	0.3759	1.0281	2.9456
	Extra Trees $p_b \rightarrow B_o$	1.3358	0.3437	1.0523	2.8757
	Random Forest $p_b \rightarrow B_o$	1.3305	0.3271	1.0523	2.7083
	ANN $p_b \rightarrow B_o$	1.3430	0.3353	1.0323	2.8333
	KNN $p_b \rightarrow B_o$	1.3499	0.3547	1.0666	2.8798
	Radius Neighbors $p_b \rightarrow B_o$	1.4102	0.0843	1.3476	2.0672
Correlaciones	Standing	1.3476	0.3938	1.0340	3.2076
	Glasø	1.3064	0.3541	1.0233	2.8431
	Al-Marhoun	1.3292	0.3260	1.0173	2.8145
	Dokla & Osman	1.3720	0.4071	0.9760	3.3041
	Petrosky & Farshad	1.3417	0.3744	1.0301	3.0003

Tabla C.4: Descripción estadística completa de los errores y R^2 para p_b

Modelo		Error relativo, [%]				Error absoluto, [psi]				R^2, [%]
		μ	d.e.	min	max	μ (MAE)	d.e.	min	max	
ML	Extra Trees	21.04	26.74	0.00	174.74	236	259	0	1272	95.24
	Random Forest	22.33	29.08	0.00	210.94	272	288	0	1362	93.93
	KNN	48.08	62.93	0.81	288.82	410	343	13	1588	88.91
	ANN	176.63	292.12	1.18	1693.33	1275	861	24	4370	67.62
	SVR	199.70	331.71	0.29	1942.97	1361	831	6	4407	8.08
	Radius Neighbors	94.91	138.68	0.08	950.45	749	522	1	2635	1.20
Correlaciones	Standing	28.12	20.51	0.64	102.92	428	495	8.93	3184	83.44
	Glasø	83.58	91.51	1.52	340.32	829	662	12.26	2826	56.34
	Al-Marhoun	25.49	23.40	1.65	153.49	493	652	12.93	3031	74.18
	Dokla & Osman	36.63	21.08	0.23	123.81	626	664	3.68	2398	67.81
	Petrosky & Farshad	84.40	175.89	0.32	1179.62	621	474	2.29	2816	76.33

Tabla C.5: Descripción estadística completa de los errores y R^2 para B_{ob}

Modelo		Error relativo, [%]				Error absoluto, [rb/STB]				R^2, [%]
		μ	d.e.	min	max	μ (MAE)	d.e.	min	max	
ML	Extra Trees	4.27	4.17	0.00	25.07	0.0567	0.0613	0.0001	0.2845	95.35
	SVR	3.66	3.87	0.04	20.18	0.0487	0.0504	0.0007	0.2096	96.72
	ANN	3.09	3.81	0.01	16.65	0.0438	0.0645	0.0001	0.3394	95.95
	Random Forest	3.11	4.34	0.08	21.71	0.0487	0.0959	0.0009	0.5455	97.45
	KNN	4.23	4.56	0.06	18.25	0.0584	0.0812	0.0007	0.5853	92.89
	Radius Neighbors	20.66	10.58	0.06	40.99	0.2786	0.2110	0.0009	1.2756	20.25
	SVR $p_b \rightarrow B_{ob}$	2.87	3.48	0.01	15.67	0.0388	0.0542	0.0001	0.2752	97.04
	Extra Trees $p_b \rightarrow B_{ob}$	2.91	3.74	0.00	16.76	0.0415	0.0640	0.0000	0.3399	96.12
	Random Forest $p_b \rightarrow B_{ob}$	3.27	4.46	0.05	19.83	0.0487	0.0897	0.0006	0.4993	93.07
	ANN $p_b \rightarrow B_{ob}$	4.03	4.33	0.07	20.14	0.0549	0.0688	0.0008	0.3931	84.92
	KNN $p_b \rightarrow B_{ob}$	3.41	4.18	0.01	18.28	0.0504	0.0788	0.0001	0.4592	82.33
	Radius Neighbors $p_b \rightarrow B_{ob}$	20.51	10.42	0.13	40.95	0.2756	0.2013	0.0018	1.1404	19.91
Correlaciones	Standing	1.94	3.86	0.00	17.28	0.0239	0.0516	0.0000	0.3159	97.85
	Glasø	3.40	2.89	0.00	13.97	0.0481	0.0541	0.0000	0.3645	96.50
	Al-Marhoun	3.30	3.80	0.05	18.10	0.0468	0.0626	0.0006	0.3931	95.92
	Dokla & Osman	4.43	4.24	0.03	20.10	0.0563	0.0517	0.0003	0.2863	96.10
	Petrosky & Farshad	2.50	3.12	0.05	15.53	0.0328	0.0417	0.0010	0.2073	98.12

Apéndice D

Datos PVT

En la tabla D.1, se muestra la descripción estadística de los datos PVT pertenecientes a yacimientos mexicanos que forman parte del conjunto de validación. Se muestra la media aritmética (μ), la desviación estándar ($d.e.$), el valor mínimo (min) y máximo (max). Mientras que en la tabla D.2, se presentan los datos PVT mencionados.

Para ambas tablas se incluyen las cuatro propiedades que sirven como datos de entrada para los algoritmos de ML ($°API$, T, γ_g y R_s). Así como las variables objetivo asociadas, es decir p_b y B_{ob}. Adicionalmente, para una mejor apreciación de las características de estos fluidos, se incluye la viscosidad del aceite a la presión de burbuja (μ_{ob}).

De acuerdo con los grados API presentados en la tabla D.2, se puede notar que los hidrocarburos van de pesados ($°API$ de 22.3° a 10.0°) a extra pesados ($°API$ menores a 10.0°). Como se menciona en el capítulo 5, los algoritmos de ML mostraron la capacidad de estimar p_b y B_{ob} con el set de datos de validación, del cual los datos mostrados en la tabla D.2 forman parte. Por lo anterior, los modelos de ML presentados en este trabajo tienen la capacidad de estimar propiedades PVT (p_b y B_{ob}) de yacimientos mexicanos con hidrocarburos pesados a extra pesados con una precisión equiparable a las correlaciones empíricas.

Figura D.1: Descripción estadística de datos PVT para yacimientos mexicanos

Parámetro	°API	T, °F	γ_g, 1	R_s, SCF/STB	μ_{ob}, cP	p_b, psi	B_{ob}, rb/STB
μ	9.55	211.9	1.1655	132.8	162.88	843	1.118
d.e.	2.03	31.8	0.1815	49.7	289.21	442	0.058
min	6.00	122.8	0.7158	50.0	12.80	261	1.000
max	15.00	251.6	1.4800	252.0	1049.82	2166	1.202

Figura D.2: Datos PVT correspondientes a yacimientos mexicanos, estos forman parte del conjunto de validación

°API	T, °F	γ_g , 1	R_{sb} , SCF/STB	μ_{ob} , cP	p_b, psi	B_{ob} , rb/STB
6.00	172.6	1.2718	81.0	775.34	479	1.078
6.70	237.2	1.2776	130.3	59.42	1027	1.129
6.70	162.3	1.4800	50.0	938.00	261	1.011
7.00	217.6	1.3470	151.0	61.00	766	1.111
8.00	217.2	0.9323	252.0	49.41	2166	1.015
8.10	181.4	1.2224	80.0	312.33	530	1.126
8.20	203.0	0.9416	236.0	70.89	2035	1.012
8.50	212.2	0.9434	80.0	80.63	713	1.138
8.60	226.4	1.3035	144.8	83.41	754	1.124
8.80	217.4	1.1240	126.0	65.95	645	1.109
9.10	172.4	0.7158	80.0	176.00	484	1.104
9.20	176.0	1.4192	70.0	1049.82	479	1.046
9.50	251.6	1.2911	163.4	42.27	796	1.164
9.50	228.2	1.2686	168.5	27.12	724	1.148
9.60	179.6	1.0929	112.0	162.11	595	1.106
9.80	122.8	1.2880	163.4	27.88	777	1.182
10.00	235.4	1.2300	116.3	37.98	701	1.149
10.00	235.4	1.2300	116.3	37.98	701	1.000
10.60	233.6	1.0221	161.0	25.10	1177	1.198
10.80	233.6	1.0318	137.0	28.54	1170	1.137
10.90	239.2	1.0010	125.2	28.21	1051	1.129
10.90	230.0	1.2970	184.7	14.45	710	1.169
12.00	205.0	0.8950	201.0	26.57	1312	1.202
12.30	230.0	1.2151	98.4	21.88	594	1.141
12.50	244.4	1.2569	120.0	19.80	683	1.186
15.00	244.4	1.2051	103.7	12.80	576	1.142

Referencias

Ahmed, T. (2018). *Reservoir engineering handbook*. Gulf professional publishing.

Al-Marhoun, M. A. (1988). PVT Correlations for Middle East Crude Oils. *Journal of Petroleum Technology*, *40*(05), 650-666. https://doi.org/10.2118/13718-PA

Al-Marhoun, M. A. (1992). New correlations for formation volume factors of oil and gas mixtures. *Journal of Canadian Petroleum Technology*, *31*(03).

American Petroleum Institute. (2003, abril). *API Recommended practice 44 second edition, Sampling Petroleum Reservoir Fluids*. API. http://www.ipt.ntnu.no/~curtis/courses/PVT-Flow/2018-TPG4145/e-notes/PVT-Papers/API-RP-44-Sampling-2nd-ed.pdf

Bandura, L., Halpert, A. D., & Zhang, Z. (2018). *Machine learning in the interpreter's toolbox: Unsupervised, supervised, and deep-learning applications* (Vol. All Days) [SEG-2018-2997015]. https://doi.org/10.1190/segam2018-2997015.1

Bánzer, C. (1996). Correlaciones numéricas PVT. *Universidad de Zulia, Maracaibo, Venezuela*.

BC Gharbi, R., & Elsharkawy, A. M. (1999). Neural network model for estimating the PVT properties of Middle East crude oils. *SPE Reservoir Evaluation & Engineering*, *2*(03), 255-265.

Beggs, H. D., & Robinson, J. R. (1975). Estimating the viscosity of crude oil systems. *Journal of Petroleum technology*, *27*(09), 1140-1141.

Brownlee, J. . (2017a, 26 de julio). *What is the Difference Between a Parameter and a Hyperparameter?* Consultado el 6 de marzo de 2022, desde https://machinelearningmastery.com/difference-between-a-parameter-and-a-hyperparameter/

Brownlee, J. . (2017b, 11 de diciembre). *Difference Between Classification and Regression in Machine Learning*. Consultado el 6 de marzo de 2022, desde https://machinelearningmastery.com/classification-versus-regression-in-machine-learning/

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., & Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 108-122.

Canipa, N. K., Galvan, C. A., Perez, J. A., & Guzman, M. A. (2003). Clasificacion de petroleos mexicanos mediante cromatografia de gases y analisis de componentes principales. *Revista de la Sociedad Quimica de Mexico*, *47*(3), 275-282.

Chouinard. (2022). k-Nearest Neighbors (KNN) in Python. https://www.jcchouinard.com/k-nearest-neighbors/

De Ghetto, G., & Villa, M. (1994). Reliability analysis on PVT correlations. *European Petroleum Conference*.

Dokla, M. E., & Osman, M. E. (1992). Correlation of PVT Properties for UAE Crudes. *SPE Formation Evaluation*, *7*(01), 41-46. https://doi.org/10.2118/20989-PA

Fernández. (s.f.). *Errores Absolutos y Relativos*. Consultado el 5 de agosto de 2022, desde https://www.fisicalab.com/apartado/errores-absoluto-relativos

Glasø, (1980). Generalized pressure-volume-temperature correlations. *Journal of Petroleum Technology*, *32*(05), 785-795.

Guillen-Rondon, P., Cobos, C., Larrazabal, G., & Diz, A. (2019). *Machine learning: A deep learning approach for seismic structural evaluation* (Vol. Day 4 Wed, September 18, 2019) [D043S152R004]. https://doi.org/10.1190/segam2019-3216712.1

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., ... Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357-362. https://doi.org/10.1038/s41586-020-2649-2

Ikiensikimama, S. S., & Ogboja, O. (2009). *New Bubblepoint Pressure Empirical PVT Correlation* (Vol. All Days) [SPE-128893-MS]. https://doi.org/10.2118/128893-MS

INEGI. (2021, 22 de junio). Comunicado de prensa núm. 352/21. https://www.inegi.org.mx/contenidos/saladeprensa/boletines/2021/OtrTemEcon/ENDUTIH_2020.pdf

Jayeola, I., Olusola, B., & Orodu, K. (2022). *Machine Learning Prediction versus Decline Curve prediction: A Niger Delta case study* (Vol. Day 2 Tue, August 02, 2022) [D021S009R002]. https://doi.org/10.2118/211956-MS

Jolliffe, I. T. (2002). *Principal component analysis*. Springer, Department of Mathematical Sciences, University of Aberdeen, Aberdeen.

Kartoatmodjo, T., & Schmidt, Z. (1994). Large data bank improves crude physical property correlations. *Oil and Gas Journal;(United States)*, *92*(27).

Khan Academy. (s.f.). *Estadística: fórmulas alternativas para la varianza*. Consultado el 5 de agosto de 2022, desde https://es.khanacademy.org/math/statistics-probability/summarizing-quantitative-data/variance-standard-deviation-population/v/statistics-alternate-variance-

formulas#:%7E:text=Para%20una%20poblaci%C3%B3n%2C%20la%20varianza, trabajar%20con%20esta%20f%C3%B3rmula%20alternativa.

Liu, Y. (2020). *Python Machine Learning by Example - Third Edition: Build Intelligent Systems Using Python, TensorFlow 2, PyTorch, and Scikit-learn*. Packt Publishing. https://books.google.com.mx/books?id=nfT-zQEACAAJ

Mal, A., Ødegård, S. I., Helgeland, S., Zulkhifly Sinaga, S., & Svendsen, M. (2022). *Prediction of Stuck Pipe Incidents Using Models Powered by Deep Learning and Machine Learning* (Vol. Day 2 Wed, March 09, 2022) [D022S005R002]. https://doi.org/10.2118/208778-MS

Mate Labs. (2017, 23 de agosto). *Secret Sauce behind the beauty of Deep Learning: Beginners guide to Activation Functions*. Consultado el 29 de junio de 2022, desde https://towardsdatascience.com/secret-sauce-behind-the-beauty-of-deep-learning-beginners-guide-to-activation-functions-a8e23a57d046#:%7E:text=Identity%20or%20Linear%20Activation%20Function, proportional%20to%20the%20input%20data.

McCain, W. (1990). *The Properties of Petroleum Fluids*. PennWell Books. https://books.google.com.mx/books?id=EWxUFzW61wkC

McKinney, W. (2010). Data Structures for Statistical Computing in Python. En S. van der Walt & J. Millman (Eds.), *Proceedings of the 9th Python in Science Conference* (pp. 56-61). https://doi.org/10.25080/Majora-92bf1922-00a

Mueller, J., & Massaron, L. (2021). *Machine Learning For Dummies*. Wiley. https://books.google.com.mx/books?id=iHtszQEACAAJ

Numbere, O. G., Azuibuike, I. I., & Ikiensikimama, S. S. (2013). *Bubble Point Pressure Prediction Model for Niger Delta Crude using Artificial Neural Network Approach* (Vol. All Days) [SPE-167586-MS]. https://doi.org/10.2118/167586-MS

Nyuytiymbiy, K. . (2020, 30 de diciembre). *Parameters and Hyperparameters in Machine Learning and Deep Learning*. Consultado el 6 de marzo de 2022, desde https://towardsdatascience.com/parameters-and-hyperparameters-aa609601a9ac

Odi, U., & Nguyen, T. (2018). *Geological Facies Prediction Using Computed Tomography in a Machine Learning and Deep Learning Environment* (Vol. Day 1 Mon, July 23, 2018) [D013S011R004]. https://doi.org/10.15530/URTEC-2018-2901881

Odutola, T. O., Bassey, I., Igbine, A., & Monday, C. U. (2022). *Hydrate Risk Management and Evaluation for Gas-Dominated Systems Using Machine Learning* (Vol. Day 2 Tue, August 02, 2022) [D021S006R005]. https://doi.org/10.2118/212000-MS

Ogwu, J., Ikpesu, E., & Ogbonna, K. (2022). *Natural Gas Spot Price Prediction Using a Machine Learning Datacentric Approach* (Vol. Day 3 Wed, August 03, 2022) [D032S003R001]. https://doi.org/10.2118/211979-MS

Omar, M., & Todd, A. (1993). Development of new modified black oil correlations for Malaysian crudes. *SPE Asia Pacific oil and gas conference*.

Onwuchekwa, C. (2018). *Application of Machine Learning Ideas to Reservoir Fluid Properties Estimation* (Vol. All Days) [SPE-193461-MS]. https://doi.org/10.2118/193461-MS

Osman, E., Abdel-Wahhab, O., & Al-Marhoun, M. (2001). *Prediction of Oil PVT Properties Using Neural Networks* (Vol. All Days) [SPE-68233-MS]. https://doi.org/10.2118/68233-MS

Palmer, C. E., & Gu, M. (2022). *Using Artificial Intelligence and Machine Learning to Assist in Completion Design of Unconventional Reservoirs* (Vol. Day 3 Wed, June 22, 2022) [D031S064R006]. https://doi.org/10.15530/urtec-2022-3723099

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825-2830.

Petrosky, G., & Farshad, F. (1993). Pressure-volume-temperature correlations for Gulf of Mexico crude oils. *SPE annual technical conference and exhibition*.

Ramirez, A. M., Valle, G. A., Romero, F., & Jaimes, M. (2017). *Prediction of PVT Properties in Crude Oil Using Machine Learning Techniques MLT* (Vol. Day 2 Thu, May 18, 2017) [D021S009R002]. https://doi.org/10.2118/185536-MS

Rogulina, A., Zaytsev, A., Ismailova, L., Kovalev, D., Katterbauer, K., & Marsala, A. (2022). *Similarity Learning for Well Logs Prediction Using Machine Learning Algorithms* (Vol. Day 3 Wed, February 23, 2022) [D032S158R005]. https://doi.org/10.2523/IPTC-22067-MS

Schlumberger. (s.f.-a). *Single-Phase Sample Bottle*. https://www.slb.com/reservoir-characterization/reservoir-testing/surface-testing/surface-sampling/single-phase-sample-bottle

Schlumberger. (s.f.-b). *Wellhead Sampling Manifold*. https://www.slb.com/reservoir-characterization/reservoir-testing/surface-testing/surface-sampling/wellhead-sampling-manifold

Shobha, G., & Rangaswamy, S. (2018). Chapter 8-Machine Learning Handbook of Statistics. Elsevier.

Standing, M. B. (1977). *Volumetric and phase behavior of oil field hydrocarbon systems*. Society of petroleum engineers of AIME.

Standing, M. (1947). A pressure-volume-temperature correlation for mixtures of California oils and gases. *Drilling and Production Practice*.

The Numpy development team. (s.f.). *Numpy*. https://github.com/numpy/numpy/blob/main/branding/logo/secondary/numpylogo2.png

The pandas development team. (s.f.). *Pandas*. https://pandas.pydata.org/about/citing.html

The scikit-learn development team. (s.f.). *Scikit-learn*. https://github.com/scikit-learn/scikit-learn/blob/main/doc/logos/scikit-learn-logo-notext.png

Ugoyah, J. C., Ajienka, J. A., Wachikwu-Elechi, V. U., & Ikiensikimama, S. S. (2022). *Prediction of Scale Precipitation by Modelling its Thermodynamic Properties using Machine Learning*

Engineering (Vol. Day 2 Tue, August 02, 2022) [D021S007R005]. https://doi.org/10.2118/212010-MS

Varotsis, N., Gaganis, V., Nighswander, J., & Guieze, P. (1999). *A Novel Non-Iterative Method for the Prediction of the PVT Behavior of Reservoir Fluids* (Vol. All Days) [SPE-56745-MS]. https://doi.org/10.2118/56745-MS

Vasquez, M., & Beggs, H. (1980). Correlations for Fluid Physical Property Prediction. *Journal of Petroleum Technology*, *32*(06), 968-970. https://doi.org/10.2118/6719-PA

Yang, X., Dindoruk, B., & Lu, L. (2020). A comparative analysis of bubble point pressure prediction using advanced machine learning algorithms and classical correlations. *Journal of Petroleum Science and Engineering*, *185*, 106598. https://doi.org/https://doi.org/10.1016/j.petrol.2019.106598

Yu. (s.f.). *Matplotlib*. https://matplotlib.org/stable/gallery/misc/logos2.html