PREPROCESAMIENTO DE DATOS EN APRENDIZAJE AUTOMÁTICO SUPERVISADO: TÉCNICAS DE NORMALIZACIÓN

By

Kelsy Pamela Cabello Solorzano

A thesis submitted in conformity with the requirements for the MSc in Economics, Finance and Computer Science

Universidad de Huelva & Universidad Internacional de Andalucía

Septiembre 2023

PREPROCESAMIENTO DE DATOS EN APRENDIZAJE AUTOMÁTICO SUPERVISADO: TÉCNICAS DE NORMALIZACIÓN

Kelsy Pamela Cabello Solorzano

Máster en Economía, Finanzas y Computación

Supervisado por:

Dr. Antonio Javier Tallon Ballesteros

Universidad de Huelva & Universidad Internacional de Andalucía

September 2023

Abstract

In Machine Learning (ML) algorithms, data normalization can play a key role. This research focuses on analyzing and comparing the impact of various normalization techniques. Five normalization techniques, Mín-Máx, Z-score, Unit Normalization, Pareto Scaling and Sigmoid, were applied as a preliminary step to the use of the eight Machine Learning algorithms (Logistic Regression, Decision Trees, Random Forest, Gradient Boosting, Support Vector Machine SVM, Neural Networks, Naïve Bayes and *K*-Nearest Neighbors). The objective of this master's thesis (TFM) is to determine in an accurate and informed way the most appropriate normalization technique for each algorithm, in order to increase the accuracy in problem solving. For the development of the research, use was made of the Python programming language, together with the Scikit-learn library package and the data analysis libraries (Pandas, Matplotlib and NumPy). Through this comparative analysis, we seek to provide reliable information to improve the performance of Machine Learning algorithms through the application of an appropriate data normalization technique.

Key words: Data Mining, Standardization Techniques, Machine learning algorithms, Mín-Máx, Zscore, Unit Normalization, Pareto Scaling and Sigmoidea.

Resumen

En los algoritmos de Machine Learning (ML), la normalización de datos puede jugar un papel fundamental. Esta investigación se centra en analizar y comparar el impacto de varias técnicas de normalización. Se aplicaron cinco técnicas de normalización: Mín-Máx, Z-score, Normalización de Unidad, Pareto Scaling y Sigmoidea, como paso previo al uso de los ocho algoritmos de Machine Learning (Regresión Logística, Árboles de decisión, Random Forest, Gradient Boosting, Support Vector Machine SVM, Redes Neuronales, Naïve Bayes y *k*-Nearest Neighbors). El objetivo de este trabajo de fin de máster (TFM) es determinar de manera precisa e informada la técnica de normalización más adecuada para cada algoritmo, con el fin de aumentar la precisión en la resolución de problemas. Para el desarrollo de la investigación se hizo uso del lenguaje de programación Python, junto con el paquete de librerías de Scikit-learn y las librerías de análisis de datos (Pandas, Matplotlib y NumPy). A través de este análisis comparativo, buscamos brindar información confiable para mejorar el desempeño de los algoritmos de Machine Learning a través de la aplicación de una adecuada técnica de normalización de datos.

Palabras claves: Minería de datos, Técnicas de estandarización, Algoritmos de aprendizaje automático, Mín-Máx, Zscore, Normalización de Unidad, Pareto Scaling y Sigmoidea

Tabla de contenidos

1. INTRODUCCIÓN 8
1.1. Objetivos de la Investigación 9
2. FUNDAMENTOS TEÓRICOS 9
2.1. Machine Learning 9
2.2. Tipos de Machine Learning 10
2.2.1. Aprendizaje Supervisado 10
2.2.2. Aprendizaje no Supervisado 10
2.2.3. Aprendizaje Semi-supervisado 11
2.2.4. Aprendizaje por Refuerzo 11
2.3. Conceptos Teóricos de Técnicas de Normalización 12
2.3.1. Mín-Máx 12
2.3.2. Z-Score 13
2.3.3. Normalización de Unidad(L2) 14
2.3.4. Pareto Scaling 14
2.3.5. Sigmoidea 15
2.4. Algoritmos de Clasificación y Regresión 15
2.4.1. Regresión Logística 15
2.4.2. Arboles de Decisión 16
2.4.3. Random Forest 16
2.4.4. Gradient Boosting 17
2.4.5. Support Vector Machine (SVM) 17
2.4.6. Redes Neuronales 18
2.4.7. Naïve Bayes 19
2.4.8. K-Nearest Neighbors (kNN) 19
2.5. La Normalización en el Proceso de Aplicación de Machine Learning 20
2.6. Métricas de Evaluación de Modelos de ML 21
3. PREPROCESAMIENTO DE DATOS 23
3.1. Selección y Adquisición del Conjunto de Datos 23
3.2. Análisis Exploratorio de Datos 24
3.3. Limpieza y Transformación de Datos 26
4. EXPERIMENTACIÓN 27
4.1. Normalización de Datos 27
4.1.1. Normalización Mín-Máx 27
4.1.2. Normalización Z-Score 32
4.1.3. Normalización de Unidad (L2) 35

4.1.4. Normalización Pareto Scaling 37
4.1.5. Normalización Sigmoidea 40
5. RESULTADOS 43
5.1. Análisis de Resultados del Datasets Antes de la Normalización 43
5.2. Análisis de Resultados del Datasets Normalizados 48
6. CONCLUSIONES 62
7. REFERENCIAS 64

Índice de Tablas

Tabla 1: Matriz de Confusión..... 22
Tabla 2: Métricas de Evaluación para Modelos de Clasificación 22
Tabla 3: Métricas de Evaluación para Modelos de Regresión 23
Tabla 4: Lista de Conjunto de Base de Datos 24
Tabla 5: Análisis Exploratorio Dataset Clasificación. 25
Tabla 6: Análisis Exploratorio Dataset Regresión 26
Tabla 7: Aplicación de la Normalización Mín-Máx [0,1] al Dataset de Clasificación. 28
Tabla 8: Aplicación de la normalización Mín-Máx al Dataset de Regresión..... 29
Tabla 9: Aplicación de la Normalización Mín-Máx [-1,1] al Dataset de Clasificación. 31
Tabla 10: Aplicación de la Normalización ZScore al Dataset de Clasificación..... 33
Tabla 11: Aplicación de la Normalización ZScore al Dataset de Regresión..... 34
Tabla 12: Aplicación de la Normalización de Unidad al Dataset de Clasificación..... 35
Tabla 13: Aplicación de la Normalización de Unidad al Dataset de Regresión..... 37
Tabla 14: Aplicación de la Normalización Pareto Scaling al Dataset de Clasificación. 37
Tabla 15: Aplicación de la Normalización de Pareto Scaling al Dataset de Regresión. 39
Tabla 16: Aplicación de la Normalización Sigmoidea al Dataset de Clasificación. 40
Tabla 17: Aplicación de la Normalización Sigmoidea al Dataset de Regresión. 41
Tabla 18: Resultado Dataset de Clasificación Antes de la Normalización. 43
Tabla 19: Resultado del Dataset Regresión Antes de la Normalización 45

Índice de figuras

Figura 1: La Normalización en el Proceso de Aplicación de Machine Learning 20
Figura 2: Resultado del Dataset Regresión Antes de la Normalización 46
Figura 3: Aplicación de Algoritmos de ML al Dataset Original - Parte 1 47
Figura 4: Aplicación de Algoritmos de ML al Dataset Original - Parte 2 47
Figura 5: Algoritmos de ML con Métricas >= 80% 49
Figura 6: Algoritmos de ML con Métricas <= 60% a 79% 50
Figura 7: Algoritmos de ML con métricas <= 59% 51
Figura 8: Normalización de Mín-Máx [0,1] & Aplicación de Algoritmo de ML - Parte 2 52
Figura 9: Normalización de Mín-Máx [0,1] & Aplicación de Algoritmo de ML- Parte 1 52
Figura 10: Normalización de Mín-Máx [-1,1] & Aplicación de Algoritmo de ML - Parte 2 53
Figura 11: Normalización de Mín-Máx [-1,1] & Aplicación de Algoritmo de ML - Parte 1 53
Figura 12: Normalización ZScore & Aplicación de Algoritmo de ML - Parte 2 54
Figura 13: Normalización ZScore & Aplicación de Algoritmo de ML- Parte 1 54
Figura 14: Normalización de Unidad & Aplicación Algoritmos de ML- Parte 2 55
Figura 15: Normalización de Unidad & Aplicación Algoritmos de ML - Parte 1 55
Figura 16: Normalización PScaling & Aplicación de Algoritmo de ML - Parte 2 56
Figura 17: Normalización PScaling & Aplicación de Algoritmo de ML - Parte 1 56
Figura 18: Normalización Sigmoidea & Aplicación de Algoritmo de ML - Parte 1 57
Figura 19: Normalización Sigmoidea & Aplicación de Algoritmo de ML - Parte 2 57
Figura 20: Algoritmos de ML con métricas aceptables - Dataset Regresión. 58
Figura 21: Algoritmos de ML con métricas Mixtos - Dataset Regresión. 58
Figura 22:Algoritmo de ML con métricas Bajos - Dataset Regresión. 59
Figura 23: Normalización Zscore & Aplicación de Algoritmo de ML – Dataset Regresión 60
Figura 24: Normalización MinMax [-1,1] & Aplicación de Algoritmo de ML–Dataset Regresión 60
Figura 25: Normalización Unidad & Aplicación de Algoritmo de ML – Dataset Regresión 60
Figura 26: Normalización MinMax [0,1] & Aplicación de Algoritmo de ML–Dataset Regresión . 60
Figura 27: Normalización Sigmoidea & Aplicación de Algoritmo de ML – Dataset Regresión..... 61
Figura 28: Normalización PScaling & Aplicación de Algoritmo de ML – Dataset Regresión..... 61

1. INTRODUCCIÓN

La normalización de datos es una etapa fundamental dentro del procesamiento de datos que se realiza antes de aplicar los algoritmos de aprendizaje automático, con el objetivo de mejorar la precisión del pronóstico al llevar las características de los datos a la misma escala y reducir los valores atípicos. Esto mejora la calidad y la coherencia de los datos y, en consecuencia, mejora la capacidad de los modelos predictivos para detectar patrones y realizar pronósticos precisos. En esta investigación, presentamos el impacto de la normalización de datos en el rendimiento de los modelos de clasificación y regresión.

En el apartado 2, presentamos los fundamentos teóricos y tipos de Machine Learning (ML), donde destacamos el aprendizaje supervisado, el aprendizaje no supervisado, el aprendizaje semi-supervisado y el aprendizaje por refuerzo. Además, resaltamos las cinco técnicas de normalización: Mín-Máx, Z-Score, Normalización de Unidad, Pareto Scaling y Sigmoidea, que se aplican en la investigación. Por otro lado, también destacamos los tipos de algoritmos de ML: Regresión Logística, Arboles de Decisión, Random Forest, Gradient Boosting, Support Vector Machine (SVM), Redes Neuronales, Naïve Bayes y *k*-Nearest Neighbors (*k*NN), que se aplican posteriormente a la normalización de datos con el fin de medir el rendimiento del modelo.

En el apartado 3, realizamos el procesamiento de datos, donde seleccionamos diez conjuntos de datos procedentes del repositorio UCI, Kaggle y OpenML conjuntos de datos de clasificación y regresión. Luego, se realiza el análisis exploratorio mediante diferentes técnicas, como histogramas, gráficos de dispersión, valores faltantes y balance de clases. Posteriormente, llevamos a cabo la limpieza y transformación de datos. En esta etapa, se pueden aplicar técnicas de transformación de datos, como la codificación de variables categóricas y la estandarización de formatos de fecha y hora en caso de que sea necesario para la investigación.

En el apartado 4, la fase de experimentación, realizamos la normalización de datos. Se aplicaron cinco técnicas de normalización: Mín-Máx, Z-Score, Normalización de Unidad, Pareto Scaling y Sigmoidea. En el caso de Mín-Máx, utilizamos dos variantes, una normalizando los valores de las características en el intervalo [0,1] y la otra normalizándolos en el intervalo [-1,1]. La experimentación se desarrolló mediante el lenguaje de programación Python, junto con el paquete de librerías de Scikit-learn y las librerías de análisis de datos (Pandas, Matplotlib y NumPy).

En el apartado 5, se realiza un análisis de los resultados obtenidos luego de aplicación de las técnicas mencionadas, y en el apartado 6, se presentan las conclusiones y

recomendaciones para futuras investigaciones. A continuación, se describen los objetivos principales de la investigación.

1.1. Objetivos de la Investigación

- **Objetivo general**

  Determinar la técnica de normalización más apropiada para cada algoritmo de Machine Learning con el fin de mejorar la precisión en resolución de problemas.

- **Objetivos específicos**

- Analizar el impacto de diversas técnicas de normalización de datos en diferentes algoritmos de Machine Learning.
- Contribuir a la comprensión de la relación entre la normalización de los datos y el rendimiento de los algoritmos de Machine Learning.

2. FUNDAMENTOS TEÓRICOS

2.1. Machine Learning

Aprendizaje automático o Machine Learning es una rama de la inteligencia artificial que tiene como objetivo desarrollar algoritmos que permitan a un sistema o programa aprender y mejorar su desempeño a partir de datos sin necesidad de ser programadas explícitamente para cada tarea.

Según el libro "Introducción to Machine Learning with Python" de Andreas Müller y Sarah Guido, se define el Machine Learning como "un conjunto de técnicas y herramientas que permiten construir modelos de datos que pueden predecir el comportamiento futuro de los datos" [1].

Asimismo, en el libro "Machine Learning: A Probabilistic Perspective" de Kevin Murphy, se describe el Machine Learning como "la ciencia de hacer que las computadoras actúen sin ser explícitamente programadas, basándose en el análisis y la inferencia de datos" [2].

2.2. Tipos de Machine Learning

2.2.1. ***Aprendizaje Supervisado***

En este tipo de aprendizaje, el sistema recibe un conjunto de datos etiquetado es decir conjunto de datos de entrada y su correspondiente resultado esperado, para aprender a relacionar las entradas con las salidas. El objetivo es crear un modelo que pueda predecir las salidas para nuevas entradas no vistas anteriormente. Ejemplos de aplicaciones de aprendizaje supervisado que incluyen la detección de spam en correo electrónico, la predicción del precio de una casa o la clasificación de imágenes en categorías.

En una fórmula general conocida se podría expresar como $y = f(x)$ donde y es el variable resultado y la x es el conjunto de información introducida al modelo. El los datos de entrenamiento (training) el x como y son conocidas, por lo tanto, esperamos que el modelo implementado aprenda la función aplicada para que cuando se introduzca nuevos conjuntos de datos se obtenga los resultados esperados.

Según el Russel R. en su Libro "*Machine Learning – Guía paso a paso para implementar algoritmos con Python*" [3] indica que este tipo de Machine Learning los datos que alimentan el algoritmo con la solución esperada son referidos como "Labels" (etiquetas) y los algoritmos a usar dentro de esta categoría son de ***clasificación*** y ***regresión***.

Los algoritmos supervisados más comunes son:

- *k*-nearest neighbors (*k*NN, *k* vecinos más cercanos).
- Red Neuronal.
- Máquinas de soporte de vectores.
- Regresión logística.
- Árboles de decisiones y bosques aleatorios.

2.2.2. ***Aprendizaje no Supervisado***

Los métodos de aprendizaje supervisado generalmente requieren de los datos de entrenamiento donde los resultados que buscamos predecir ya están disponibles o es decir etiquetados como discretos o con valores continuos. Sin embargo, existen situaciones que no tenemos la libertad o la ventaja de tener datos para el entrenamiento pre etiquetados y necesitamos de métodos para extraer información o patrones útiles de nuestros datos. Son escenarios perfectos donde actúan los métodos de aprendizaje no supervisado, como su

nombre lo indica “no supervisado” porque el modelo o algoritmos intenta aprender estructuras, patrones y relaciones a partir de los datos de entrada sin ninguna ayuda o supervisión con el fin de extraer ideas o información significativas de los datos. Los métodos de aprendizaje no supervisado se pueden clasificar en las siguientes tareas de Machine Learning más relevantes par aprendizaje no supervisado [4].

- Clustering (*k* medias, análisis de agrupamiento jerárquico)
- Reducción de la dimensionalidad
- Detección de anomalías
- Minería de reglas de asociación.

En resumen, en este tipo de aprendizaje, el sistema recibe un conjunto de datos sin etiquetar y tiene como objetivo encontrar patrones o estructuras en los datos. No hay resultados esperados, y el modelo tiene que descubrir por sí mismos lo que es importante en los datos. Por ejemplo, la aplicación de este tipo de Machine Learning incluyen segmentación de clientes en grupo similares, la detección de anomalías de los datos y la reducción de la dimensión de los datos.

2.2.3. ***Aprendizaje Semi-supervisado***

Este método de aprendizaje suele caer entre los métodos de aprendizaje supervisado y no supervisado. Es decir, usan una gran cantidad de datos de entrenamiento sin etiquetas (componente de aprendizaje no supervisado) y una pequeña cantidad de datos pre etiquetados y anotados (componente de aprendizaje supervisado). Para entender explicamos mediante una analogía por ejemplo se quiere entender y traducir al texto. Para comenzar sería difícil usar una forma no supervisada debido a que no se sabe que significan esas palabras y para usar un sistema supervisado se necesita indicar cuales son todos los sonidos de las palabras para que el modelo aprenda a diferenciarlas. Al usar una forma semi-supervisada se una data de palabras comunes y frases, comenzando con una clasificación básica. En esta parte se usa la forma supervisada, luego se alimenta el modelo con una nueva data de forma no supervisada para que sea analizado.

2.2.4. ***Aprendizaje por Refuerzo***

Este tipo de aprendizaje hace que el sistema interactúa con un ambiente en el que tiene que tomar decisiones para maximizar una recompensa o minimizar los costos. El

sistema recibe una retroalimentación sobre las acciones tomadas y utiliza esa información para mejorar su desempeño. Los pasos principales de un método de aprendizaje por refuerzo se mencionan a continuación: Preparar al agente con el conjunto de políticas iniciales y estrategias, Observa el entorno y el estado actual, Selecciona la directiva optima y realice la acción, Obtenga la recompensa correspondiente, Actualice las políticas si es necesarios y Repita los pasos de forma iterativa hasta que el agente aprenda las políticas más optimas.

Algunos ejemplos de la aplicación de aprendizaje por refuerzo incluyen los juegos de mesa, la navegación de robots autónomos, la gestión de carteras de inversión, etc.

2.3. Conceptos Teóricos de Técnicas de Normalización

La normalización tiene como objetivo principal, garantizar la calidad de los datos antes de aplicar cualquier algoritmo de Machine Learning, el resultado que se obtiene indica la efectividad del método de normalización aplicada [5] . Una de las ventajas es que se puede usar para escalar los datos en el mismo rango de valores por cada característica de entrada con el fin de minimizar el sesgo dentro de un modelo de Machine Learning de una u otra característica. Además, puede agilizar el tiempo de entrenamiento para cada característica dentro de la misma escala y especialmente es aplicable para modelos que tienen como entrada escalas desiguales.

A continuación, se explican las diferentes técnicas de normalización y sus diferentes reglas para su correcta aplicación.

*2.3.1. **Mín-Máx***

La técnica de normalización Mín-Máx, también conocida como escalado, es un método de normalización de datos muy utilizado en el aprendizaje automático. Su principal objetivo es transformar los valores de los datos para que caigan dentro de un rango determinado, normalmente entre 0 y 1 de manera que el valor minimo se transforma en 0 y el valor maximo se transforma en 1.

El proceso de normalización Mín-Máx implica restar el valor mínimo del conjunto de datos y luego dividirlo por la diferencia entre los valores máximo y mínimo. Esto se hace para cada valor del conjunto de datos. Para cada atributo, el Mín-Máx de una entrada que se calcula mediante:

$$x^{(i)}{}_{norm} = \frac{(x^{(i)} - x_{min})}{(x_{max} - x_{min})} \quad (1)$$

donde x es el valor original del atributo, en la entrada i del conjunto de datos, y x_{min} y x_{max} son los valores mínimo y máximo del atributo en el conjunto de datos.

La normalización Mín-Máx es útil cuando se desea conservar información sobre los datos y ajustarlos a un rango específico. Al escalar los datos dentro de un rango estrecho, la comparación y el análisis de las variables resultan más sencillos, especialmente cuando se utilizan algoritmos sensibles al tamaño de los datos, como las redes neuronales. Además del rango de 0 a 1, la técnica Mín-Máx puede ajustar los datos a otros rangos definidos por el usuario y requeridos por el problema. Por ejemplo, la fórmula Mín-Máx puede ajustarse aún más si se requiere un rango entre -1 y 1. Es importante señalar que la normalización Mín-Máx transforma los datos a una escala determinada, pero no cambia la forma de la distribución de los valores originales. Sin embargo, este método es sensible a los sesgos, ya que afectan al rango de la escala y comprimen la mayoría de los valores en un rango muy estrecho.

2.3.2. ***Z-Score***

La técnica de puntuación Z, también conocida como normalización, es un método de normalización de datos muy utilizado en el aprendizaje automático. Su principal objetivo es transformar los datos para que tengan una media de cero y una desviación estándar de uno. El proceso de normalización Z-Score consiste en restar la media de los datos y dividirla por la desviación estándar. Esto se hace para cada valor del conjunto de datos. Para cada atributo, la puntuación Z de una entrada se calcula mediante:

$$z^{(i)} = \frac{x^{(i)} - \mu}{\sigma} \quad (2)$$

donde μ es la media del conjunto de datos y σ es la desviación estándar del conjunto de datos.

La normalización de la puntuación Z es útil cuando se desea comparar y utilizar variables con diferentes escalas y distribuciones. Al normalizar los datos, todas las variables tienen un tamaño y una distribución comparables, lo que facilita la comparación y el análisis. Con la normalización de la puntuación Z, los valores por encima de la media tendrán una puntuación Z positiva, mientras que los valores por debajo de la media tendrán

una puntuación Z negativa. Esto permite identificar valores atípicos o valores extremos porque tienden a tener una puntuación Z alta o baja en comparación con el resto de los datos. La normalización de la puntuación Z no transforma los datos dentro de un rango determinado, sino que los ajusta para que tengan una media cero y una desviación típica de 1. Además, la técnica asume que los datos siguen una distribución normal y, por lo tanto, funciona bien para variables continuas.

2.3.3. ***Normalización de Unidad(L2)***

La técnica de normalización unitaria, también conocida como normalización L2, es una técnica común utilizada en el aprendizaje automático para normalizar datos. Su objetivo es garantizar que los vectores propios tengan una longitud unitaria euclidiana. El proceso de normalización unitaria se realiza dividiendo el valor de cada vector propio por la norma euclídea del vector original.

$$x_{un}^{(i)} = \frac{x^{(i)}}{\sqrt{\sum_j x^{(j)^2}}} \tag{3}$$

al dividir cada valor por la norma euclídea, la norma del vector es igual a 1. Esta técnica es especialmente útil cuando se utilizan algoritmos sensibles al tamaño de los datos, como los modelos basados en la distancia, en los que el tamaño de la característica afecta a la similitud o la distancia entre instancias. La normalización de los datos a una escala única impide que las características con valores grandes dominen la contribución del modelo. Es importante señalar que esta técnica no transforma los datos en un área específica, sino que los ajusta a la escala general, lo que preserva la escala relativa.

2.3.4. ***Pareto Scaling***

Esta técnica es similar a la técnica de escalado automático en lugar de desviación estándar, la técnica de Pareto Scaling utiliza la raíz cuadrada de la deviación estándar como factor de escala [6]. La normalización de Pareto Scaling se aplica con el objetivo de transformar las variables numéricas a un rango especifico, el objetivo de la esta técnica es calcular la media y la desviación estándar para luego buscar ajustar cada valor mediante la siguiente formula:

$$x' = \frac{(x - \min(x))}{\sqrt{(var(x))}} \tag{4}$$

2.3.5. ***Sigmoidea***

Es una técnica de normalización que utiliza la función sigmoide para ajustar los valores de una variable en un rango en específico. La función sigmoide es una técnica matemática que toma valor real como entrada y procede un valor en el rango (0,1) como salida [5]. El objetivo de es útil cuando se desea ajustar los valores de una variable en específico y controlar su escala. Esta técnica puede ayudar a evitar que los valores extremadamente grandes o pequeños afecten en el análisis o normalización.

$$x_{norm} = \frac{1}{1-e^{-x}} \tag{5}$$

Es importante señalar que esta técnica puede cambiar la distribución y la interpretación de los datos originales, esto debido a que utiliza la función no lineal para ajustar los valores. Por los tanto es recomendable evaluar el impacto de esta técnica en el rendimiento del modelo según las características específicas del conjunto de datos.

2.4. Algoritmos de Clasificación y Regresión

2.4.1. ***Regresión Logística***

Es un algoritmo de aprendizaje supervisado utilizado en Machine Learning para la solución de problemas de clasificación binaria. El concepto básico de regresión logística es predecir la probabilidad de que una instancia pertenezca a una clase determinada. La salida que se espera del algoritmo es un valor entre 0 y 1, esto representa la probabilidad estimada. Con esta probabilidad obtenida se puede establecer un umbral para asignar una clase especifica a la instancia.

Según la investigación [7] resalta que la regresión logística es un modelo de clasificación para la estimación de funciones que mide la relación entre variables independientes y una variable dependiente categórica, y mediante la aproximación de la función de densidad probabilística condicional utilizando la función logística que también se conoce como sigmoidal.

El algoritmo de regresión logística utiliza una función logística, también conocida como función sigmoide, para modelar la relación entre las variables de entrada (características) y la variable de salida (clase). La función sigmoide tiene la forma matemática:

$$f(x) = \frac{1}{1-e^{-x}} \tag{6}$$

donde x es una combinación lineal de las características de entrada ponderadas por los coeficientes del modelo. La función sigmoide mapea el resultado de esta combinación lineal a un valor entre 0 y 1.

2.4.2. ***Arboles de Decisión***

El árbol de decisión es un algoritmo de aprendizaje automático ampliamente utilizado en problemas de clasificación y regresión. Se basa en un árbol compuesto por nodos y ramas, donde cada rama representa una característica o atributo del conjunto de datos y cada nodo representa una regla de decisión basada en ese atributo. El árbol comienza con un nodo raíz que abarca todo el conjunto de datos y se divide en nodos secundarios a medida que se desciende por el árbol, utilizando criterios como la ganancia de información o la reducción de la impureza.

En la clasificación, las hojas del árbol representan las clases a las que pertenecen los datos, mientras que en la regresión representan valores numéricos. Durante el entrenamiento, el árbol se ajusta a los datos dividiendo los nodos de manera recursiva para minimizar el error. Los árboles de decisión son populares en Machine Learning debido a su facilidad de implementación, interpretación y capacidad para manejar diferentes tipos de datos.

Según la investigacion [8] un árbol de clasificación se construye a través de la división de su nodo principal en 2 nodos internos. En cada nodo el algoritmo busca en todas las variables una a una para obtener la mejor división. El punto de partida para la división se indica mediante un nodo raíz t en la parte superior del árbol y la división se decide por una condición c para un solo variable x, $x \leq c$.

2.4.3. ***Random Forest***

Es un modelo que fue desarrollado por Leo Breiman [9], es un grupo de árboles de clasificación o regresión sin podar a partir de la selección aleatoria de muestras de los datos de entrenamiento. Las características son seleccionados en el proceso de inducción, la predicción se hace agregando (voto mayoritario para la clasificación o promedio para la regresión) las predicciones del conjunto.

Los bosques aleatorios se basan en un conjunto de árboles de decisión con ideas de agrupación y Bootstrap. Además, en considerado como un método estadístico no

paramétrico muy bueno para abordar problemas de regresión, clasificación binaria y clasificación multiclase en un marco único y versátil. desde el punto de vista práctico los bosques aleatorios son ampliamente más usados y presentan resultados impactantes con solo unos pocos parámetros para ajustar [10].

2.4.4. ***Gradient Boosting***

Es un algoritmo con mayor impacto para resolver problemas de regresión y clasificación. Se basa en combinar múltiples modelos de aprendizaje débil y convertirse en un modelo fuerte.

El objetivo de Gradiente Boosting es construir un modelo aditivo, donde se agregan sucesivamente varios modelos débiles, conocidos como arboles de decisión para mejorar la precisión general del modelo. Los árboles se construyen de manera secuencial para corregir los errores del modelo anterior, estos modelos se miden mediante la funcion de perdida. El algoritmo Gradient Boosting utiliza el gradiente descendente para minimizar la funcion de perdida. En cada iteración, se calcula el gradiente de la función de pérdida con respecto a las predicciones actuales del modelo y se ajusta el siguiente modelo débil en la dirección opuesta al gradiente, de modo que la función de pérdida se minimice. Esto permite que el modelo se enfoque en las áreas en las que tiene dificultades para hacer predicciones precisas. Una vez que se han agregado todos los modelos débiles, se obtiene un modelo fuerte combinando sus predicciones ponderadas. La ponderación de cada modelo débil se determina en función de su contribución relativa a la reducción de la función de pérdida.

Leo Breiman impulsó que se puede interpretar la gradiente como un algoritmo de aumento en una funcion de costo adecuado[11]. Años más tarde las investigaciones de Jerome H. Friedman desarrollo algoritmos de aumento de gradiente de regresión explícitos y el aumento de gradiente funcional más general que fue presentado por Llew Mason, Jonathan Baxter y Marcus Frean [12].

2.4.5. ***Support Vector Machine (SVM)***

Es uno de los algoritmos de aprendizaje supervisado que se utiliza tanto para problemas de clasificación y regresión. Según la investigacion titulada "Improvement of Support Vector Machine Algorithm in Big Data Background" su mecanismo de trabajo es encontrar un hiperplano adecuado para segmentar las muestras de datos recopiladas. El

principio de segmentación es maximizar el intervalo (incluido el intervalo duro y el intervalo suave) y finalizarlo en un problema especial de programación cuadrática para resolver. Además, las SVM tienen como base la idea de resolver problemas de alta dimensión, utilizando numeros reducidos de vectores de soporte para minimizar el riesgo estructural. Estas máquinas son muy efectivos para la creciente diversidad y dimensionalidad de los datos, los algoritmos de SVM se destacan por su capacidad de manejar datos de grandes volúmenes sin limitaciones en los valores de los atributos, lo que las convierte en una de las herramientas más sólidas para la construcción de modelos predictivos dentro de este contexto [13].

En la investigacion [14] resalta que SVM es una técnica discriminante que tiene como objetivo encontrar una función discriminante que pueda predecir correctamente la etiqueta para una instancia recién recibida en función de un conjunto de datos de entrenamiento independiente e igualmente distribuido. A diferencia de los enfoques de aprendizaje automático generativo, que requieren el cálculo de distribución de probabilidad condicional, una función de clasificación discriminante toma un punto de datos x y lo asigna a una de varias clases que forman parte de un problema de clasificación.

2.4.6. ***Redes Neuronales***

Las redes neuronales son modelos de aprendizaje automático inspirado en el funcionamiento del cerebro humano. Estos modelos son ampliamente utilizados como herramientas de clasificación de datos por su capacidad para aprender y reconocer patrones complejos en un complejo de datos. Es un modelo computacional basado en un conjunto de unidades neuronales simples, cada uno tiene con una condición diferente y esta se denomina “funcion de activación”, esta debe cumplir antes de propagarse la informacion a través del resto de unidades neuronales. Cada unidad neuronal está conectada con otras a través de enlaces asociados a pesos que activan, aumentan, disminuyen o desactivan el valor de cada neurona de la red [15].

En la clasificación de datos mediante redes neuronales involucra un proceso de entrenamiento en que la red se alimenta de un conjunto de datos de entrenamiento previamente etiquetados. En la fase de entrenamiento, la red ajusta los pesos de las conexiones entre las neuronas para minimizar la diferencia entre las salidas de la red y las etiquetas conocidas en los datos de entrenamiento. Una vez entrenada la red neuronal esta apto para clasificar nuevos datos, los datos se propagan a través de la red. El procesamiento

se realiza mediante cálculos en cada neurona y se obtiene la salida que representa la clasificación o predicción de la predicción de la red.

2.4.7. ***Naïve Bayes***

Son un conjunto de algoritmos de aprendizaje supervisado basados en la aplicación del teorema de Bayes, con precisión de clasificación competitiva y eficiencia computacional, es decir, el tiempo de entrenamiento es lineal, características que hacen que Bayes sea muy conocido en la industria [16].

La investigación [17] “Emotion recognition based on the speech, using a Naïve Bayes Classifier” afirma concluye que este clasificador mostró resultados superiores durante su implementación en aplicaciones del mundo real. Se destaca su rapidez, buen rendimiento y sofisticación como método de clasificación. Las principales ventajas de este clasificador incluyen el supuesto de independencia condicional, que contribuye a una clasificación más ágil, y el uso de hipótesis probabilísticas para obtener resultados que representan la probabilidad de pertenecer a cada clase.

Además, el artículo [18] "Naïve Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation" destaca que Naïve Bayes (NB) es reconocido como un algoritmo de clasificación probabilística. A pesar de su simplicidad, este algoritmo demuestra ser eficiente y posee una amplia gama de aplicaciones en el mundo real. Entre las áreas destacadas se encuentran el reconocimiento de productos, el diagnóstico médico y el control de vehículos autónomos esto resalta que Naïve Bayes es conocido por su excelente rendimiento.

2.4.8. ***K-Nearest Neighbors (kNN)***

Es un algoritmo de aprendizaje automático para la solucion de problemas de clasificación y la regresión. *k*NN se basa en la suposición de que las muestras de la misma clase deben estar más cercanos al espacio de características. Para una muestra con clase desconocida, *k*NN calcula la distancia entre esta muestra de prueba y todas las muestras de entrenamiento y se les asigna la clase determina por los K vecinos más cercanos de esta prueba. El valor de K es un hiper parámetro que se establece antes de aplicar el algoritmo [19].

En la investigación [20] "*k*NN Model-Based approach in classification" resalta que *k*NN es un método de clasificación no paramétrico. El algoritmo recupera a los vecinos más cercanos de un registro de datos para clasificarlos y usa la votación mayoritaria entre los registros de datos. El éxito de este algoritmo depende del valor apropiado de *k*. Existen muchas formas de seleccionar el valor de *k*, pero la más simple es ejecutar el algoritmo varias veces con diferentes valores de *k* y elegir el que tenga el mejor rendimiento.

2.5. La Normalización en el Proceso de Aplicación de Machine Learning

Al igual que una investigación científica la aplicación de Machine Learning (ML) hace uso de datos por lo tanto requiere un proceso o una serie de pasos para llevar a cabo una correcta aplicación de las técnicas con el objetivo de lograr los resultados esperados. A continuación, se describen los pasos generales para la aplicación de ML en una investigación que requiere de la normalización de datos:

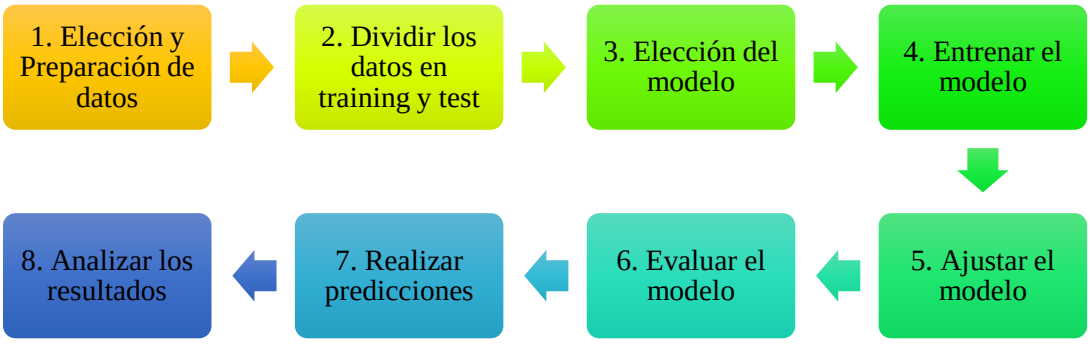


Figura 1: La Normalización en el Proceso de Aplicación de Machine Learning

1. **Elección y Preparación de datos:** Implica la elección del conjunto de datos para luego limpiar, eliminar los valores atípicos, valores faltantes y transformar los datos si es necesario.

   En este proceso, se considera especialmente la normalización de datos, ya que esta práctica resulta fundamental para identificar y eliminar valores atípicos que puedan distorsionar los resultados del modelo. La presencia de valores atípicos puede ejercer un sesgo negativo en la calidad del modelo, de ahí la importancia de su eliminación previa a la creación del mismo. Además, la normalización de datos contribuye a garantizar que los datos se encuentren en un formato uniforme y específico, lo que facilita su análisis y asegura su precisión y consistencia

2. **Dividir los datos en training y test**: Los datos se dividen en conjunto de entrenamiento que se utiliza para entrenar el modelo y el conjunto de prueba se utiliza para evaluar el rendimiento del modelo.
3. **Elección del modelo:** Elegir el modelo automático que se utilizara para hacer la predicción. La elección del modelo dependerá del tipo de datos y del problema que se aborda. En esta investigacion elegimos modelos de clasificación y de recesión.
4. **Entrenar el modelo:** Se utiliza el conjunto de datos de entrenamiento para entrenar el modelo seleccionado.
5. **Ajustar el modelo para obtener el mejor rendimiento posible:** Esto puede implicar ajustar los parámetros del modelo o elegir diferentes características
6. **Evaluar el modelo** utilizando el conjunto de datos de prueba. Se puede utilizar métricas de evaluación como: accuracy, precisión, sensibilidad, F1 score, error cuadrático (MSE), error absoluto medio (MAE), raíz del error cuadrático (RMSE) y R^2.
7. **Realizar predicciones:** Se utiliza datos de entrada para hacer predicciones utilizando el modelo entrenado
8. **Analizar los resultados:** se analizan las predicciones y se determina si el modelo es lo suficientemente preciso para su uso en situaciones reales.

2.6. Métricas de Evaluación de Modelos de ML

Son herramientas que permiten medir el rendimiento de los modelos de Machine Learning, estas métricas ayudan a determinar si el resultado es aceptado o requiere mayor evaluación.

Es un paso esencial en todo proyecto de Machine Learning (ML) y las métricas de evaluación de modelos se utilizan para evaluar el ajuste entre la salida del modelo y los datos [21]. Además, permiten la comparación entre diferentes modelos para seleccionar el más adecuado para la tarea o el problema que se está tratando de resolver [22].

- **Matriz de Confusión**

La matriz de confusión se usa para evaluar el rendimiento del modelo donde la salida puede ser de dos o más clases. Esta métrica de evaluación muestra a través de una tabla los tipos de errores que genera el modelo. La salida de la matriz de confusión corresponde a una matriz de dimensión N x N, en donde N es el número de clases predichas y se realiza la comparación entre las predicciones obtenidas con los datos reales

[23]. Por ejemplo, se tiene 2 clases A y B, el objetivo principal de la métrica es determinar el número de muestra de la clase A que fueron clasificados dentro del grupo de la clase B. La composición de la matriz de confusión es:

Tabla 1: Matriz de Confusión

Clase	**Positivo**	**Negativo**
Positivo	Verdadero Positivo (TP)	Falso Negativo (FN)
Negativo	Falso Positivo (FP)	Verdadero Negativo (TN)

Fuente Artículo: A Two-Step Data Normalization Approach for Improving Classification Accuracy in the Medical Diagnosis Domain [24]

Existen diferentes métricas de evaluación para modelos de Machine Learning (ML) pero los más comunes son:

Tabla 2: Métricas de Evaluación para Modelos de Clasificación

Métricas para Modelos de Clasificación		
Métrica	**Definición**	**Expresión**
Accuracy	Es el porcentaje de los resultados que el modelo predice correctamente en el total de predicciones. La precisión optima es el 100%.	$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$
Precisión	La precisión es la relación entre las predicciones correctas positivas y los resultados previstos positivos generales.	$Precision = \frac{TP}{(TP + FP)}$
Sensibilidad (Recall)	Es la proporción de predicciones positivas verdaderas en comparación con todos los datos positivos verdaderos.	$\text{Sensitivity} = \frac{TP}{(TP + FN)}$
F1 Score	Es una matriz de confusión que tiene en cuenta la relación entre precisión y sensibilidad.	$F1 - score = 2\,\frac{Precision\ x\ \text{Recall}}{Precision + \ \text{Recall}}$
ROC curve	Es una representación gráfica del rendimiento de un modelo de clasificación en función de diferentes umbrales de decisión. No tiene una fórmula única, sino que se crea trazando la tasa de verdaderos positivos (sensibilidad) en función de la tasa de falsos positivos para varios umbrales.	

Fuente: *"Comparison of Support Vector Machine and K-Nearest Neighbors in Breast Cancer Classification" and "Homogeneous data normalization and deep learning: A case study in human activity classification"* [23], [25]

Tabla 3: Métricas de Evaluación para Modelos de Regresión

Métricas para Modelos de Regresión		
Métrica	**Definición**	**Expresión**
Error cuadrático medio (MSE)	Es la suma de los cuadrados de los errores entre las predicciones y los valores reales. Cuanto menor sea el MSE será mejor el modelo.	$MSE = (Y - \hat{Y})$ ^2 Y = valor real $\hat{Y} = prediccion$
Error absoluto medio (MAE)	Es la suma de los valores absolutos de los errores entre las predicciones y valores reales. Cuanto menor sea el MAE, será mejor el modelo.	$MAE = \lvert Y - \hat{Y} \rvert$
Raíz del error cuadrático medio (RMSE)	Es la raíz cuadrada del MSE. Es una medida de la dispersión de los errores. Cuanto menor sea el RMSE, mejor será el modelo.	$RMSE = \sqrt{(MSE)}$
$\boldsymbol{R^2}$	Es un coeficiente de determinación. Mide la proporción de la varianza de los valores reales que se explica por las predicciones del modelo. Cuanto mayor sea el R^2, será mejor el modelo.	$R^2 = 1 - \left(\frac{SSres}{SStot}\right)$ SSres = suma de los cuadrados de los residuos SStot = suma de cuadrados totales

3. PREPROCESAMIENTO DE DATOS

El preprocesamiento de datos es uno de los pasos más críticos en un proceso de minería de datos, debido a que busca mejorar la eficiencia y la facilidad de identificar el conjunto de datos para una investigación en específico. Además, se ocupa de la preparación y transformación de la versión inicial del conjunto de datos. Los métodos de preprocesamiento de datos incluyen los pasos más comunes como: selección, análisis exploratorio, limpieza y transformación de datos.

3.1. Selección y Adquisición del Conjunto de Datos

En esta sección se describe los 10 conjuntos de datos fueron extraídas del repositorio de la Universidad de California en Irvine (UCI) [26], Kaggle [27] y OpenML

[28]. La descripción detalla de los conjuntos de datos se encuentran en las referencias de cada conjunto de datos.

Tabla 4: Lista de Conjunto de Base de Datos

Id	Nombre de la base de datos	Instancias	N°.de atributos	Target	Clases	Problema	Referencia
01	Madelon Dataset	2600	500	Class	2	Clasificación	[29]
02	Boston house price Dataset	506	14	Medv	2	Regresión	[30]
03	Body fat percentage estimates Dataset	252	15	class	2	Regresión	[31]
04	Heart Attack Analysis & Prediction Dataset	303	14	Class	2	Clasificación	[32]
05	Heart Failure Prediction Dataset	299	13	Class	2	Clasificación	[33]
06	Breast Cancer Wisconsin Dataset	699	10	Class	2	Clasificación	[34]
07	Elevators Dataset	16599	19	Class	2	Clasificación	[35]
08	Fried Dataset	40768	11	Class	2	Clasificación	[36]
09	Puma32H Dataset	8192	33	Class	2	Clasificación	[37]
10	Tumours of the central nervous system Dataset	60	7130	Class	2	Clasificación	[38]

3.2. Análisis Exploratorio de Datos

En la fase de análisis exploratorio se realizó un análisis detallado de la calidad de datos, con el objetivo de comprender su estructura, característica y relaciones. Se utilizó diferentes técnicas estadísticas para la visualización de datos, identificar patrones, tendencias, valores atípicos y la distribución de los datos.

- **Histograma**

El histograma es una técnica que muestra el comportamiento de los variables de un conjunto de datos, en el eje X horizontal se representan las barras con el rango de valores de cada variable, además indica la cantidad de veces que un valor es representado. En el eje Y se representa la frecuencia o cantidad de ocurrencias por cada rango de valores. En resumen, para la normalización se espera que las barras del histograma de las variables del conjunto de datos tengan un comportamiento simétrico, dando referencia que los conjuntos de datos tienen una distribución normal.

- **Gráficos de Dispersión**

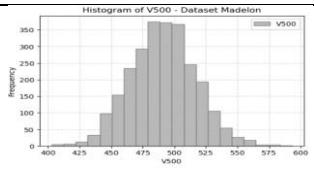
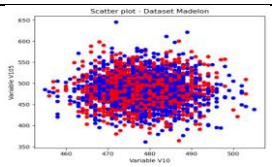
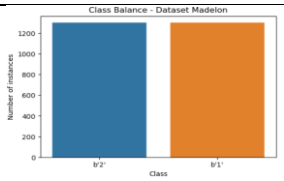
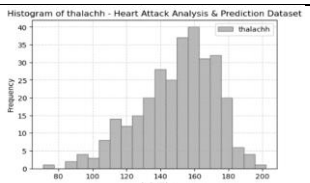
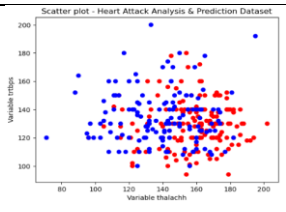
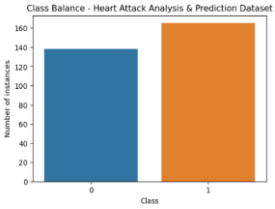
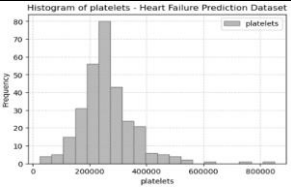
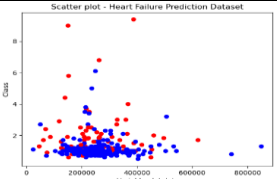
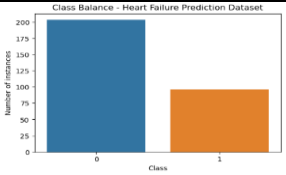
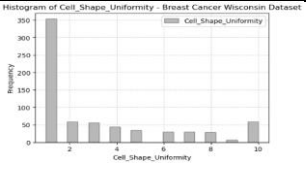
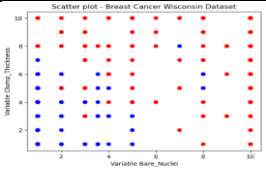
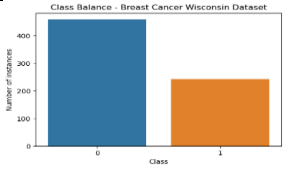
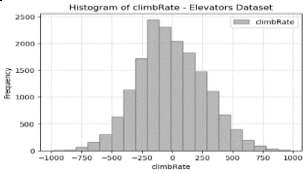
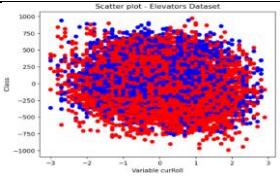
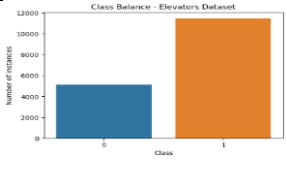
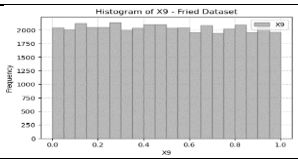
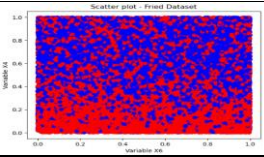
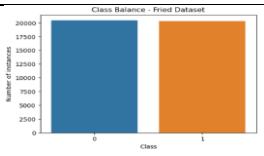
Es una representación gráfica de puntos o scatter plot de un conjunto de datos en un sistema de coordenadas, cada punto representa a un par de valores correspondientes a dos variables. Las variables independientes por lo general se ubican en el eje horizontal (eje

X), mientras que la variable independiente se coloca en el eje vertical (eje Y). Para la normalización de datos los grafios de dispersión ayudan a identificar patrones, tendencias, correlaciones y valores atípicos.

- **Balance de Clases**

Es una técnica que permite visualizar en gráficos de barras la distribución de las clases de un conjunto de datos. Examina la cantidad de instancias que tienen en cada clase y verifica si hay un desequilibrio significativo entre ellas. Esto ayudará a tomar decisiones informadas sobre el manejo de desequilibrios en la fase de entrenamiento de los modelos de ML.

Tabla 5: Análisis Exploratorio Dataset Clasificación.

N°	Nombre de la Base de Datos	Histograma	Gráficos de Dispersión	Balance de Clases
01	Madelon Dataset			
04	Heart Attack Analysis & Prediction Dataset			
05	Heart Failure Prediction Dataset			
06	Breast Cancer Wisconsin Dataset			
07	Elevators Dataset			
08	Fried Dataset			

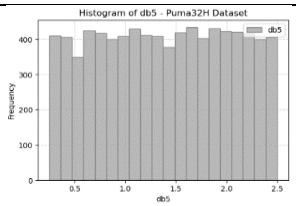
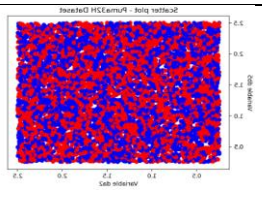
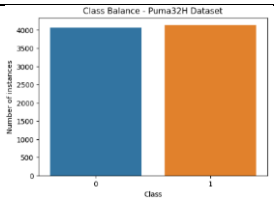
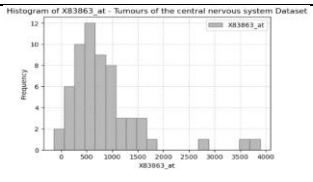
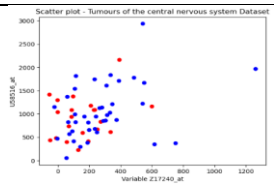
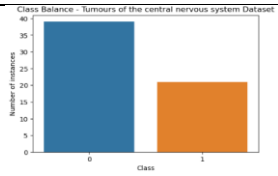
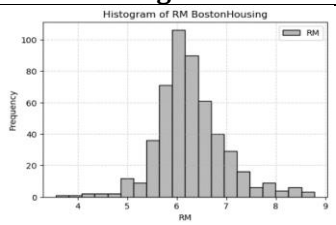
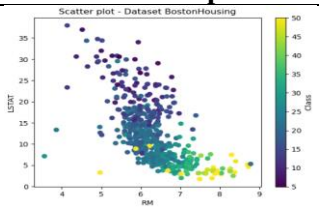
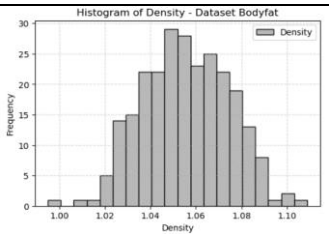
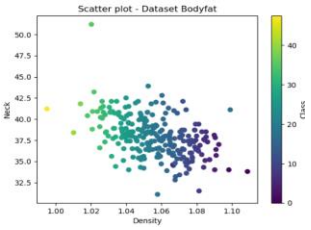
09	Puma32H Dataset			
10	Tumours of the central nervous system Dataset			

Tabla 6: Análisis Exploratorio Dataset Regresión

N°	Nombre de la base de datos	Histograma	Gráficos de Dispersión
02	Boston house price Dataset		
03	Body fat percentage estimates Dataset		

3.3. Limpieza y Transformación de Datos

En esta etapa se realiza diferentes acciones para limpiar el conjunto de dato para garantizar su calidad, el manejo de valores faltantes, la eliminación de datos duplicados y la correlación de errores. Además, se puede aplicar técnicas de transformación de datos, como la codificación de variables categóricos y la estandarización de formatos fecha y hora. En los 10 conjuntos de datos seleccionados para esta investigación algunos requerían de la aplicación de las técnicas de codificación One-hot Ecoding mientras que otros no requerían ninguna técnica de transformación, esto debido a que los conjuntos de datos tenían valores numéricos y esto facilita la aplicación de las técnicas de normalización.

4. EXPERIMENTACIÓN

Para la normalización de los conjuntos de datos seleccionados se aplicaron las 5 técnicas de normalización (Mín-Máx, Zscore, Normalización de Unidad, Pareto Scaling y Sigmoidea) con la herramienta de Python.

Python: Lenguaje de programación que contiene una variedad de librerías y paquetes especializados como: numpy, pandas, matplolib, scipy y Scikit-Learn que facilitan la manipulación y la transformación, además son útiles para cálculos numéricos, estadísticos, la visualización de datos y la facilidad para la implementación de algoritmos de aprendizaje automático.

4.1. Normalización de Datos

El proceso de normalización de datos es una etapa fundamental dentro del procesamiento de datos en Machine Learning. Consiste en transformar los datos en un rango, escala común o en una distribución especifica con el objetivo de garantizar que las características tengan una escala similar y sigan una distribución deseada a fin de minimizar el sesgo al aplicar los modelos de Machine Learning. Este paso es importante porque muchos algoritmos de Machine Learning asumen que los datos están normalizados o tienen una distribución especifica.

Según la investigación [5] "Impact of Data Normalization on Stock Index" resalta que la normalización busca garantizar la calidad de datos antes de introducir cualquier tipo de algoritmo de aprendizaje y además puede acelerar el tiempo de entrenamiento al iniciar el proceso de entrenamiento para todas las funciones de la misma escala.

Es importante recalcar que la normalización es recomendable realizar antes de dividir los datos en train y test, para evitar filtrar información del conjunto de prueba en el proceso de normalización y evitar un sesgo en la evaluación del modelo.

4.1.1. Normalización Mín-Máx

Esta técnica de normalización transforma los datos a un rango específico, de [0, 1] o [-1, 1]. Esto preserva la proporción relativa de los datos y es útil cuando se requiere mantener las relaciones de orden. Además, es un método de normalización de datos muy utilizado en el aprendizaje automático. En la siguiente Tabla 7 se muestra el resultado la aplicación de Mín-Máx [0,1] en el conjunto de datos de clasificación.

Tabla 7: Aplicación de la Normalización Mín-Máx [0,1] al Dataset de Clasificación.

Dataset	Classifier	Mín-Máx [0,1]					
		% Precisión	**Exactitud**	**Precisión**	**Sensibilidad**	**F1-score**	**AUC**
01. Madelon Dataset	1. Regresión Logística	54.04	0.54	0.53	0.57	0.55	0.54
	2. Árboles de decisión	76.54	0.77	0.74	0.80	0.77	0.77
	3. Random Forest	74.62	0.75	0.77	0.70	0.73	0.75
	4. Gradient Boosting	77.31	0.77	0.78	0.75	0.77	0.77
	5. SVM	54.23	0.54	0.54	0.56	0.55	0.54
	6. Redes Neuronales	57.12	0.57	0.59	0.43	0.50	0.57
	7. Naïve Bayes	59.62	0.60	0.59	0.58	0.58	0.60
	8. *k*NN	58.08	0.58	0.59	0.50	0.54	0.58
04. Heart Attack Análisis & Prediction Dataset	1. Regresión Logística	85.25	0.85	0.90	0.81	0.85	0.85
	2. Árboles de decisión	83.61	0.84	0.84	0.84	0.84	0.84
	3. Random Forest	85.25	0.85	0.85	0.88	0.86	0.85
	4. Gradient Boosting	78.69	0.79	0.81	0.78	0.79	0.79
	5. SVM	85.25	0.85	0.87	0.84	0.86	0.85
	6. Redes Neuronales	81.97	0.82	0.86	0.78	0.82	0.82
	7. Naïve Bayes	86.89	0.87	0.90	0.84	0.87	0.87
	8. *k*NN	81.97	0.82	0.86	0.78	0.82	0.82
05. Heart Failure Prediction Dataset	1. Regresión Logística	71.67	0.72	0.90	0.36	0.51	0.67
	2. Árboles de decisión	73.33	0.73	0.70	0.64	0.67	0.72
	3. Random Forest	76.67	0.77	0.87	0.52	0.65	0.73
	4. Gradient Boosting	75.00	0.75	0.78	0.56	0.65	0.72
	5. SVM	75.00	0.75	0.92	0.44	0.59	0.71
	6. Redes Neuronales	71.67	0.72	0.72	0.52	0.60	0.69
	7. Naïve Bayes	70.00	0.70	0.82	0.36	0.50	0.65
	8. *k*NN	58.33	0.58	0.50	0.08	0.14	0.51
06. Breast Cáncer Wisconsin Dataset	1. Regresión Logística	97.14	0.97	0.98	0.93	0.95	0.96
	2. Árboles de decisión	96.43	0.96	0.95	0.93	0.94	0.96
	3. Random Forest	97.14	0.97	0.96	0.96	0.96	0.97
	4. Gradient Boosting	95.71	0.96	0.95	0.91	0.93	0.95
	5. SVM	97.14	0.97	0.98	0.93	0.95	0.96
	6. Redes Neuronales	96.43	0.96	0.98	0.91	0.94	0.95
	7. Naïve Bayes	96.43	0.96	0.92	0.98	0.95	0.97
	8. *k*NN	98.57	0.99	0.98	0.98	0.98	0.98
07. Elevators Dataset	1. Regresión Logística	86.78	0.87	0.85	0.98	0.91	0.80
	2. Árboles de decisión	80.30	0.80	0.80	0.95	0.87	0.72
	3. Random Forest	85.60	0.86	0.85	0.95	0.90	0.80
	4. Gradient Boosting	84.70	0.85	0.84	0.96	0.90	0.87
	5. SVM	87.62	0.88	0.85	0.99	0.92	0.81
	6. Redes Neuronales	89.55	0.90	0.89	0.96	0.93	0.86
	7. Naïve Bayes	75.27	0.75	0.74	0.98	0.84	0.92
	8. *k*NN	81.20	0.81	0.82	0.93	0.87	0.74
08. Fried Dataset	1. Regresión Logística	83.90	0.84	0.84	0.84	0.84	0.84
	2. Árboles de decisión	82.14	0.82	0.85	0.78	0.81	0.82
	3. Random Forest	91.45	0.91	0.92	0.91	0.91	0.91
	4. Gradient Boosting	91.67	0.92	0.91	0.92	0.92	0.92
	5. SVM	84.17	0.84	0.85	0.84	0.84	0.84
	6. Redes Neuronales	88.30	0.88	0.88	0.89	0.88	0.88
	7. Naïve Bayes	86.77	0.87	0.86	0.88	0.87	0.87
	8. *k*NN	86.60	0.87	0.86	0.87	0.87	0.87
09. Puma32H Dataset	1. Regresión Logística	64.86	0.65	0.66	0.66	0.66	0.65
	2. Árboles de decisión	87.55	0.88	0.89	0.86	0.88	0.88
	3. Random Forest	86.88	0.87	0.88	0.87	0.87	0.87
	4. Gradient Boosting	87.43	0.87	0.89	0.87	0.88	0.87

	5. SVM	65.22	0.65	0.66	0.67	0.67	0.65
	6. Redes Neuronales	86.21	0.86	0.85	0.90	0.87	0.86
	7. Naïve Bayes	64.98	0.65	0.66	0.68	0.67	0.65
	8. *k*NN	59.79	0.60	0.62	0.58	0.60	0.60
10. Tumours of the central nervous system Dataset	1. Regresión Logística	66.67	0.67	0.50	0.25	0.33	0.56
	2. Árboles de decisión	58.33	0.58	0.33	0.25	0.29	0.50
	3. Random Forest	75.00	0.75	1.00	0.25	0.40	0.62
	4. Gradient Boosting	66.67	0.67	0.50	0.50	0.50	0.62
	5. SVM	58.33	0.58	0.33	0.25	0.29	0.50
	6. Redes Neuronales	66.67	0.67	0.00	0.00	0.00	0.50
	7. Naïve Bayes	75.00	0.75	0.60	0.75	0.67	0.75
	8. *k*NN	50.00	0.50	0.33	0.50	0.40	0.50

Analizando la Tabla 7 de resultados de la aplicación de la técnica de normalización Mín-Máx [0,1], se observa que los algoritmos con mejores resultados son los Árboles de Decisión y el Gradient Boosting en el dataset N°01 el Random Forest y SVM en el dataset N°04 y nuevamente el Random Forest y Gradient Boosting en el dataset N°05. Para el conjunto N°06, las Redes Neuronales y *k*NN son los algoritmos destacados en términos de rendimiento global. Estos algoritmos demuestran la capacidad de equilibrar la precisión y la sensibilidad, lo que es crucial para futuras investigaciones.

Finalmente, la curva ROC es un gráfico de la tasa de verdaderos positivos (TPR) frente a la tasa de falsos positivos (FPR). Cuanto mayor sea la curva ROC, mejor será el rendimiento del algoritmo. Las curvas ROC de los Árboles de Decisión y del Bosque Aleatorio también son las más altas de todos los conjuntos de datos. Por lo tanto, los Árboles de Decisión y el Bosque Aleatorio son los algoritmos de mejor rendimiento en general.

En la siguiente Tabla 8 se muestra el resultado de la aplicación Mín-Máx en ambas escalas [0,1] y [-1,1] a los conjuntos de datos de regresión.

Tabla 8: Aplicación de la normalización Mín-Máx al Dataset de Regresión.

Dataset	**Classifier**	**Mín-Máx [0,1]**				**Mín-Máx [-1,1]**			
		MSE	**MAE**	**RMSE**	**R^2**	**MSE**	**MAE**	**RMSE**	R^2
02. Boston house price Dataset	1. Regresión Lineal	24.82	3.31	4.98	0.66	24.82	3.31	4.98	0.66
	2. Regresor Árbol de Decisión	11.92	2.57	3.45	0.84	10.80	2.47	3.29	0.85
	3. Random Forest	8.92	2.13	2.99	0.88	9.86	2.16	3.14	0.87
	4. Gradient Boosting	6.39	1.95	2.53	0.91	6.52	1.96	2.55	0.91
	5. SVM	37.18	3.61	6.10	0.49	37.18	3.61	6.10	0.49
	6. Redes Neuronales	25.27	3.13	5.03	0.66	22.46	3.14	4.74	0.69
	7. *k*NN	35.52	3.22	5.96	0.52	35.52	3.22	5.96	0.52

03. Body fat porcentaje estimates Dataset	1. Regresión Lineal	0.38	0.46	0.62	0.99	0.38	0.46	0.62	0.99
	2. Regresor Árbol de Decisión	0.39	0.35	0.62	0.99	1.92	0.51	1.38	0.96
	3. Random Forest	0.05	0.16	0.23	1.00	0.05	0.16	0.22	1
	4. Gradient Boosting	0.08	0.21	0.29	1.00	0.09	0.21	0.29	1
	5. SVM	8.46	1.95	2.91	0.82	8.46	1.95	2.91	0.82
	6. Redes Neuronales	30.47	4.75	5.52	0.35	3.13	1.51	1.77	0.93
	7. *k*NN	7.75	2.22	2.78	0.83	7.75	2.22	2.78	0.83

En la Tabla 8, las métricas de evaluación de los modelos de regresión indican que el algoritmo de Regresión Lineal presenta un MSE de 24.82 y un R^2 por debajo de 0.66 en ambas escalas [0,1] y [-1,1] utilizando la técnica Mín-Máx. Esto sugiere que el modelo no se ajusta bien a este tipo de conjuntos de datos, como el N°02. Por otro lado, el algoritmo del Árbol de Decisión muestra valores de MSE de 11.92 y 10.8, con un R^2 de 0.84 y 0.85 respectivamente. El algoritmo Random Forest presenta MSE de 8.92 y 9.86, con R^2 de 0.88 y 0.87. El algoritmo Gradient Boosting muestra un MSE de 6.39 y 6.52, con un R^2 de 0.91. Estos resultados indican que los modelos se adaptan bien al conjunto de datos y sugieren que a mayor valor de R^2, el modelo es de mejor calidad.

De manera similar, el algoritmo SVM muestra un MSE de 37.18 y un R^2 de 0.49. El algoritmo *k*NN presenta un MSE de 35.52 y un R^2 de 0.52. Por su parte, el algoritmo de Redes Neuronales muestra MSE de 25.27 y 22.46, junto con un R^2 de 5.03 y 0.69 respectivamente. Estos resultados sugieren que los modelos no son apropiados para este tipo de conjunto de datos, como el N°02, y se recomienda analizar las variables y/o aplicar otras técnicas de normalización para lograr mejores resultados. En el dataset N°03, los algoritmos con mejores resultados son Random Forest, con un MSE de 0.05 y R^2 de 1, y Gradient Boosting, con un MSE de 0.08 y R^2 de 1. Estos dos algoritmos muestran valores mucho más altos, lo que indica que son los modelos más sólidos.

Los algoritmos de Regresión Lineal y Árbol de Decisión presentan valores de MSE de 0.38 y 1.92, con R^2 de 0.99 y 0.96 respectivamente. Ambos algoritmos muestran resultados aceptables. Entre el Support Vector Machine (SVM) y *k*NN, se observa que ambos algoritmos tienen valores de MSE y R^2 similares, oscilando entre un MSE de 7.75 y 8.46, y un R^2 de 0.82 a 0.83 en ambas escalas utilizando la técnica Mín-Máx.

Finalmente, se observa que el algoritmo de Redes Neuronales muestra valores poco favorables, con un MSE de 30.47 y un R^2 de 0.35 en la escala Mín-Máx [0,1], y un MSE de 3.13 y un R^2 de 0.93 en la escala Mín-Máx [-1,1]. Esto sugiere que el algoritmo no se

adapta bien a la escala Mín-Máx [0,1] y, por lo tanto, no obtiene buenos resultados en ese caso.

La Tabla 9 muestra la aplicación de la técnica de normalización Mín-Máx en su escala de [-1, 1] a los conjuntos de datos de clasificación.

Tabla 9: Aplicación de la Normalización Mín-Máx [-1,1] al Dataset de Clasificación.

Dataset	Classifier	Mín-Máx [-1,1]					
		% Precisión	**Exactitud**	**Precisión**	**Sensibilidad**	**F1-score**	**AUC**
01. Madelon Dataset	1. Regresión Logística	54.42	0.54	0.54	0.58	0.56	0.54
	2. Árboles de decisión	75.58	0.76	0.75	0.76	0.76	0.76
	3. Random Forest	74.62	0.75	0.76	0.71	0.73	0.75
	4. Gradient Boosting	77.12	0.77	0.78	0.75	0.76	0.77
	5. SVM	55.77	0.56	0.55	0.60	0.57	0.56
	6. Redes Neuronales	57.50	0.57	0.57	0.58	0.58	0.58
	7. Naïve Bayes	59.62	0.6	0.59	0.58	0.58	0.6
	8. *k*NN	58.08	0.58	0.59	0.50	0.54	0.58
04. Heart Attack Análisis & Prediction Dataset	1. Regresión Logística	83.61	0.84	0.87	0.81	0.84	0.84
	2. Árboles de decisión	81.97	0.82	0.82	0.84	0.83	0.82
	3. Random Forest	85.25	0.85	0.85	0.88	0.85	0.85
	4. Gradient Boosting	78.69	0.79	0.81	0.78	0.79	0.79
	5. SVM	86.89	0.87	0.88	0.88	0.88	0.87
	6. Redes Neuronales	86.89	0.87	0.85	0.91	0.88	0.87
	7. Naïve Bayes	86.89	0.87	0.90	0.84	0.87	0.87
	8. *k*NN	81.97	0.82	0.86	0.78	0.82	0.82
05. Heart Failure Prediction Dataset	1. Regresión Logística	76.67	0.77	0.92	0.48	0.63	0.73
	2. Árboles de decisión	73.33	0.73	0.70	0.64	0.67	0.72
	3. Random Forest	75.00	0.75	0.81	0.52	0.63	0.72
	4. Gradient Boosting	73.33	0.73	0.76	0.52	0.62	0.70
	5. SVM	81.67	0.82	0.94	0.60	0.73	0.79
	6. Redes Neuronales	76.67	0.77	0.82	0.56	0.67	0.74
	7. Naïve Bayes	70.00	0.70	0.82	0.36	0.50	0.65
	8. *k*NN	58.33	0.58	0.50	0.08	0.14	0.51
06. Breast Cáncer Wisconsin Dataset	1. Regresión Logística	96.43	0.96	0.98	0.91	0.94	0.95
	2. Árboles de decisión	95.00	0.95	0.95	0.89	0.92	0.93
	3. Random Forest	97.14	0.97	0.98	0.93	0.95	0.96
	4. Gradient Boosting	95.71	0.96	0.95	0.91	0.93	0.95
	5. SVM	97.14	0.97	0.98	0.93	0.95	0.96
	6. Redes Neuronales	96.43	0.96	0.95	0.93	0.94	0.96
	7. Naïve Bayes	96.43	0.96	0.92	0.98	0.95	0.97
	8. *k*NN	98.57	0.99	0.98	0.98	0.98	0.98
07. Elevators Dataset	1. Regresión Logística	89.10	0.89	0.88	0.98	0.92	0.84
	2. Árboles de decisión	79.52	0.80	0.85	0.85	0.85	0.76
	3. Random Forest	85.27	0.85	0.85	0.95	0.90	0.80
	4. Gradient Boosting	84.70	0.85	0.84	0.96	0.90	0.78
	5. SVM	89.25	0.89	0.87	0.98	0.93	0.84
	6. Redes Neuronales	89.94	0.90	0.90	0.96	0.93	0.87
	7. Naïve Bayes	75.27	0.75	0.74	0.98	0.84	0.62
	8. *k*NN	81.20	0.81	0.82	0.93	0.87	0.74
Fried Datas	1. Regresión Logística	83.91	0.84	0.84	0.84	0.84	0.84

	2. Árboles de decisión	87.33	0.87	0.87	0.88	0.87	0.87
	3. Random Forest	91.76	0.92	0.92	0.92	0.92	0.92
	4. Gradient Boosting	91.67	0.92	0.91	0.92	0.92	0.92
	5. SVM	84.16	0.84	0.84	0.84	0.84	0.84
	6. Redes Neuronales	93.17	0.93	0.93	0.94	0.93	0.93
	7. Naïve Bayes	86.77	0.87	0.86	0.88	0.87	0.87
	8. *k*NN	86.60	0.87	0.86	0.87	0.87	0.87
09. Puma32H Dataset	1. Regresión Logística	64.86	0.65	0.66	0.66	0.66	0.65
	2. Árboles de decisión	84.75	0.85	0.85	0.86	0.85	0.85
	3. Random Forest	91.76	0.92	0.92	0.92	0.92	0.92
	4. Gradient Boosting	87.43	0.87	0.89	0.87	0.88	0.87
	5. SVM	65.22	0.65	0.66	0.67	0.67	0.65
	6. Redes Neuronales	86.58	0.87	0.88	0.85	0.87	0.87
	7. Naïve Bayes	64.98	0.65	0.66	0.68	0.67	0.65
	8. *k*NN	59.79	0.60	0.62	0.58	0.60	0.60
10. Tumours of the central nervous system Dataset	1. Regresión Logística	66.67	0.67	0.50	0.25	0.33	0.56
	2. Árboles de decisión	66.67	0.67	0.50	0.50	0.50	0.62
	3. Random Forest	66.67	0.67	0.50	0.25	0.33	0.56
	4. Gradient Boosting	66.67	0.67	0.50	0.50	0.50	0.62
	5. SVM	58.33	0.58	0.33	0.25	0.29	0.50
	6. Redes Neuronales	66.67	0.67	0.00	0.00	0.00	0.50
	7. Naïve Bayes	75.00	0.75	0.60	0.75	0.67	0.75
	8. *k*NN	50.00	0.50	0.33	0.50	0.40	0.50

Según la Tabla 9, el resultado de la aplicación de algoritmos de clasificación (ML) al conjunto de datos, que fue normalizado con Mín-Máx [-1, 1], muestra que los algoritmos que tienden a tener un rendimiento sólido en la mayoría de los 8 conjuntos de datos son Random Forest, Gradient Boosting y SVM. Por otro lado, los algoritmos de Redes Neuronales y Regresión Logística obtienen resultados razonables, pero varían en diferentes conjuntos de datos. Finalmente, el algoritmo Naïve Bayes y K-Nearest Neighbors tienen resultados mixtos, con una variabilidad considerable en su rendimiento en diferentes conjuntos de datos.

*4.1.2. **Normalización Z-Score***

La técnica de normalización Zscore, transforma los datos de modo que tengan la media en cero y desviación estándar a uno. Esto hace que los datos tengan una distribución normal estándar. A continuación, en la Tabla 10 se muestra la aplicación de la técnica Zscore al conjunto de datos de Clasificación.

Tabla 10: Aplicación de la Normalización ZScore al Dataset de Clasificación.

Dataset	Classifier	Zscore					
		% Precisión	Exactitud	Precisión	Sensibilidad	F1-score	AUC
01. Madelon Dataset	1. Regresión Logística	52.69	0.53	0.52	0.57	0.54	0.53
	2. Árboles de decisión	76.35	0.76	0.74	0.80	0.77	0.76
	3. Random Forest	70.38	0.70	0.71	0.68	0.69	0.70
	4. Gradient Boosting	77.31	0.77	0.78	0.75	0.77	0.77
	5. SVM	55.58	0.56	0.55	0.60	0.57	0.56
	6. Redes Neuronales	55.19	0.55	0.55	0.54	0.55	0.55
	7. Naïve Bayes	59.62	0.60	0.59	0.58	0.58	0.60
	8. *k*NN	57.50	0.57	0.58	0.50	0.54	0.57
04. Heart Attack Análisis & Prediction Dataset	1. Regresión Logística	85.25	0.85	0.87	0.84	0.86	0.85
	2. Árboles de decisión	81.97	0.82	0.84	0.81	0.83	0.82
	3. Random Forest	88.52	0.89	0.88	0.91	0.89	0.88
	4. Gradient Boosting	78.69	0.79	0.81	0.78	0.79	0.79
	5. SVM	86.89	0.87	0.88	0.88	0.88	0.87
	6. Redes Neuronales	80.33	0.80	0.86	0.75	0.80	0.81
	7. Naïve Bayes	86.89	0.87	0.80	0.84	0.87	0.87
	8. *k*NN	91.80	0.92	0.94	0.91	0.92	0.92
05. Heart Failure Prediction Dataset	1. Regresión Logística	80.00	0.80	0.93	0.56	0.70	0.77
	2. Árboles de decisión	73.33	0.73	0.70	0.64	0.67	0.72
	3. Random Forest	76.67	0.77	0.87	0.52	0.65	0.73
	4. Gradient Boosting	73.33	0.73	0.76	0.52	0.62	0.70
	5. SVM	80.00	0.80	0.88	0.60	0.71	0.77
	6. Redes Neuronales	71.67	0.72	0.70	0.56	0.62	0.69
	7. Naïve Bayes	70.00	0.70	0.82	0.36	0.50	0.65
	8. *k*NN	70.00	0.70	1.00	0.28	0.44	0.64
06. Breast Cáncer Wisconsin Dataset	1. Regresión Logística	96.43	0.96	0.98	0.91	0.94	0.95
	2. Árboles de decisión	96.43	0.96	0.95	0.93	0.94	0.96
	3. Random Forest	97.86	0.98	0.98	0.96	0.97	0.97
	4. Gradient Boosting	95.71	0.96	0.95	0.91	0.93	0.95
	5. SVM	97.14	0.97	0.98	0.93	0.95	0.96
	6. Redes Neuronales	97.14	0.97	0.98	0.93	0.95	0.96
	7. Naïve Bayes	96.43	0.96	0.92	0.98	0.95	0.97
	8. *k*NN	97.14	0.97	0.98	0.93	0.95	0.96
07. Elevators Dataset	1. Regresión Logística	89.61	0.90	0.89	0.96	0.93	0.86
	2. Árboles de decisión	80.30	0.80	0.80	0.95	0.87	0.72
	3. Random Forest	84.76	0.85	0.85	0.95	0.89	0.79
	4. Gradient Boosting	84.70	0.85	0.84	0.96	0.90	0.78
	5. SVM	89.52	0.90	0.88	0.97	0.93	0.85
	6. Redes Neuronales	90.33	0.90	0.91	0.96	0.93	0.87
	7. Naïve Bayes	73.92	0.74	0.73	0.99	0.84	0.59
	8. *k*NN	81.14	0.81	0.82	0.93	0.87	0.75
08. Fried Dataset	1. Regresión Logística	83.91	0.84	0.84	0.84	0.84	0.84
	2. Árboles de decisión	82.14	0.82	0.85	0.78	0.81	0.82
	3. Random Forest	91.50	0.92	0.91	0.91	0.91	0.92
	4. Gradient Boosting	91.67	0.92	0.91	0.92	0.92	0.92
	5. SVM	84.16	0.84	0.84	0.84	0.84	0.84
	6. Redes Neuronales	93.40	0.93	0.93	0.94	0.93	0.93
	7. Naïve Bayes	86.77	0.87	0.86	0.88	0.87	0.87
	8. *k*NN	86.63	0.87	0.86	0.87	0.87	0.87
09. Puma32H Dataset	1. Regresión Logística	64.86	0.65	0.66	0.66	0.66	0.65
	2. Árboles de decisión	87.68	0.88	0.90	0.86	0.88	0.88
	3. Random Forest	87.61	0.88	0.89	0.87	0.88	0.88
	4. Gradient Boosting	87.43	0.87	0.89	0.87	0.88	0.87
	5. SVM	65.16	0.65	0.66	0.67	0.67	0.65
	6. Redes Neuronales	88.10	0.88	0.89	0.88	0.88	0.88

	7. Naïve Bayes	64.98	0.65	0.66	0.68	0.67	0.65
	8. *k*NN	59.24	0.59	0.62	0.57	0.59	0.59
10. Tumours of the central nervous system Dataset	1. Regresión Logística	58.33	0.58	0.33	0.25	0.29	0.50
	2. Árboles de decisión	58.33	0.58	0.40	0.50	0.44	0.56
	3. Random Forest	66.67	0.67	0.50	0.25	0.33	0.56
	4. Gradient Boosting	66.67	0.67	0.50	0.50	0.50	0.62
	5. SVM	58.33	0.58	0.33	0.25	0.29	0.50
	6. Redes Neuronales	50.00	0.50	0.38	0.75	0.50	0.56
	7. Naïve Bayes	75.00	0.75	0.60	0.75	0.67	0.75
	8. *k*NN	41.67	0.42	0.29	0.50	0.36	0.44

Según el análisis de los resultados de la aplicación de la técnica de normalización Z-score a los 8 conjuntos de datos de clasificación (Tabla 10), los valores indican que, de los 8 algoritmos aplicados, el algoritmo con mejor rendimiento general en términos de accuracy, precisión, recall, f1-score y ROC curve es Random Forest. Además, se observa que los algoritmos Regresión Logística, Árboles de decisión, SVM, Redes Neuronales y *k*NN también tienen resultados aceptables. Por otro lado, los algoritmos que tienen valores muy bajos son Naïve Bayes y Gradient Boosting. Esto significa que el rendimiento de los algoritmos varía según el conjunto de datos y es importante realizar más entrenamiento del modelo, probar otros algoritmos y analizar los recursos computacionales.

A continuación, en la Tabla 11 se muestra la aplicación de la técnica Zscore al conjunto de datos de regresión.

Tabla 11: Aplicación de la Normalización ZScore al Dataset de Regresión.

Dataset	**Classifier**	**Zscore**			
		MSE	**MAE**	**RMSE**	R^2
02. Boston house price Dataset	1. Regresión Lineal	24.82	3.31	4.98	0.66
	2. Regresor Árbol de Decisión	10.88	2.57	3.3	0.85
	3. Random Forest	9.21	2.15	3.03	0.87
	4. Gradient Boosting	6.64	1.97	2.58	0.91
	5. SVM	30.8	3.07	5.55	0.58
	6. Redes Neuronales	17.72	2.89	4.21	0.76
	7. *k*NN	28.35	2.85	5.32	0.61
03. Body fat porcentaje estimates Dataset	1. Regresión Lineal	0.38	0.46	0.62	0.99
	2. Regresor Árbol de Decisión	0.6	0.37	0.77	0.99
	3. Random Forest	0.05	0.16	0.22	1
	4. Gradient Boosting	0.08	0.2	0.28	1
	5. SVM	10.23	2.1	3.2	0.78
	6. Redes Neuronales	21.74	3.14	4.66	0.53
	7. *k*NN	9.33	2.49	3.05	0.8

Según los resultados presentados en la Tabla 11, los algoritmos con un desempeño superior en ambos conjuntos de datos son Random Forest y Gradient Boosting. Estos algoritmos han logrado valores de MSE por debajo de 0.99, lo cual indica que el modelo presenta una distribución de errores más precisa. Por otro lado, los restantes algoritmos como la Regresión Lineal, el Regresor de Árbol de Decisión, Gradient Boosting, *k*NN y SVM, exhiben resultados mixtos en ambos conjuntos de datos. Esto implica que los valores varían en función del conjunto de datos, oscilando entre niveles aceptables y otros menos satisfactorios.

*4.1.3. **Normalización de Unidad (L2)***

La normalización de unidad (L2): Divide cada muestra de datos por su norma L2, respectivamente. Esto asegura que las muestras tengan una norma unitaria, lo que puede ser útil en ciertos algoritmos que dependen de la magnitud de los datos. Los resultados de la aplicación se encuentran en la siguiente Tabla 12.

Tabla 12: Aplicación de la Normalización de Unidad al Dataset de Clasificación.

Dataset	Classifier	Normalización de Unidad					
		% Precisión	**Exactitud**	**Precisión**	**Sensibilidad**	**F1-score**	**AUC**
01. Madelon Dataset	1. Regresión Logística	52.69	0.53	0.52	0.57	0.54	0.53
	2. Árboles de decisión	76.73	0.77	0.74	0.81	0.77	0.77
	3. Random Forest	71.15	0.71	0.73	0.67	0.70	0.71
	4. Gradient Boosting	77.31	0.77	0.78	0.75	0.77	0.77
	5. SVM	55.58	0.57	0.56	0.55	0.56	0.55
	6. Redes Neuronales	55.58	0.56	0.55	0.54	0.55	0.56
	7. Naïve Bayes	59.62	0.60	0.59	0.58	0.58	0.60
	8. *k*NN	57.50	0.57	0.58	0.50	0.54	0.57
04. Heart Attack Análisis & Prediction Dataset	1. Regresión Logística	85.25	0.85	0.87	0.84	0.86	0.85
	2. Árboles de decisión	81.97	0.82	0.84	0.81	0.83	0.82
	3. Random Forest	83.61	0.84	0.84	0.84	0.84	0.84
	4. Gradient Boosting	78.69	0.79	0.81	0.78	0.79	0.79
	5. SVM	86.89	0.87	0.88	0.88	0.88	0.87
	6. Redes Neuronales	80.33	0.80	0.86	0.75	0.80	0.81
	7. Naïve Bayes	86.89	0.87	0.90	0.84	0.87	0.87
	8. *k*NN	91.80	0.92	0.94	0.91	0.92	0.92
05. Heart Failure Prediction Dataset	1. Regresión Logística	81.67	0.82	0.94	0.60	0.73	0.79
	2. Árboles de decisión	76.67	0.77	0.72	0.72	0.72	0.76
	3. Random Forest	73.33	0.73	0.70	0.64	0.67	0.72
	4. Gradient Boosting	75.00	0.75	0.81	0.52	0.63	0.72
	5. SVM	73.33	0.73	0.76	0.52	0.62	0.70
	6. Redes Neuronales	80.00	0.80	0.88	0.60	0.71	0.77
	7. Naïve Bayes	71.67	0.72	0.70	0.56	0.62	0.69
	8. *k*NN	70.00	0.70	1.00	0.28	0.44	0.64
06. Breast Cáncer Wisconsin Dataset	1. Regresión Logística	96.43	0.96	0.98	0.91	0.94	0.95
	2. Árboles de decisión	96.43	0.96	0.95	0.93	0.94	0.96
	3. Random Forest	97.14	0.97	0.98	0.93	0.95	0.96

	4. Gradient Boosting	95.71	0.96	0.95	0.91	0.93	0.95
	5. SVM	97.14	0.97	0.98	0.93	0.95	0.96
	6. Redes Neuronales	97.14	0.97	0.98	0.93	0.95	0.96
	7. Naïve Bayes	96.43	0.96	0.92	0.98	0.95	0.97
	8. *k*NN	97.14	0.97	0.98	0.93	0.95	0.96
07. Elevators Dataset	1. Regresión Logística	89.61	0.90	0.89	0.96	0.93	0.86
	2. Árboles de decisión	80.30	0.80	0.80	0.95	0.87	0.72
	3. Random Forest	85.57	0.86	0.85	0.95	0.90	0.80
	4. Gradient Boosting	84.70	0.85	0.84	0.96	0.90	0.78
	5. SVM	89.52	0.90	0.88	0.97	0.93	0.85
	6. Redes Neuronales	90.42	0.90	0.91	0.96	0.93	0.87
	7. Naïve Bayes	73.92	0.74	0.73	0.99	0.84	0.59
	8. *k*NN	81.14	0.81	0.82	0.93	0.87	0.75
08. Fried Dataset	1. Regresión Logística	83.91	0.84	0.84	0.84	0.84	0.84
	2. Árboles de decisión	82.14	0.82	0.85	0.78	0.81	0.82
	3. Random Forest	91.76	0.92	0.92	0.91	0.92	0.92
	4. Gradient Boosting	91.67	0.92	0.91	0.92	0.92	0.92
	5. SVM	84.16	0.84	0.84	0.84	0.84	0.84
	6. Redes Neuronales	93.38	0.93	0.93	0.94	0.93	0.93
	7. Naïve Bayes	86.77	0.87	0.86	0.88	0.87	0.87
	8. *k*NN	86.63	0.87	0.86	0.87	0.87	0.87
09. Puma32H Dataset	1. Regresión Logística	64.86	0.65	0.66	0.66	0.66	0.65
	2. Árboles de decisión	87.55	0.88	0.89	0.86	0.88	0.88
	3. Random Forest	87.25	0.87	0.88	0.88	0.88	0.87
	4. Gradient Boosting	87.43	0.87	0.89	0.87	0.88	0.87
	5. SVM	65.16	0.65	0.66	0.67	0.67	0.65
	6. Redes Neuronales	88.04	0.88	0.88	0.88	0.88	0.88
	7. Naïve Bayes	64.98	0.65	0.66	0.68	0.67	0.65
	8. *k*NN	59.24	0.59	0.62	0.57	0.59	0.59
10. Tumours of the central nervous system Dataset	1. Regresión Logística	58.33	0.58	0.33	0.25	0.29	0.50
	2. Árboles de decisión	66.67	0.67	0.50	0.50	0.50	0.62
	3. Random Forest	58.33	0.58	0.00	0.00	0.00	0.44
	4. Gradient Boosting	66.67	0.67	0.50	0.50	0.50	0.62
	5. SVM	58.33	0.58	0.33	0.25	0.29	0.50
	6. Redes Neuronales	58.33	0.58	0.43	0.75	0.55	0.62
	7. Naïve Bayes	75.00	0.75	0.60	0.75	0.67	0.75
	8. *k*NN	41.67	0.42	0.29	0.50	0.36	0.44

De acuerdo a la Tabla 12, los algoritmos que presentaron los mejores resultados en general fueron las Redes Neuronales, las cuales lograron una exactitud superior al 80% en 6 datasets. Tanto la Regresión Logística como los Árboles de Decisión y el Random Forest obtuvieron resultados por encima del 80% en 5 datasets cada uno. Por otro lado, Gradient Boosting, SVM y *k*NN mostraron resultados mixtos, ya que en 4 datasets alcanzaron una exactitud superior al 80%, pero en otros casos obtuvieron valores por debajo del umbral esperado. Finalmente, el Naïve Bayes obtuvo una exactitud por encima del 80% en 3 datasets, mientras que en otros 3 datasets logró una exactitud oscilante por encima del 70%, y en 2 datasets logró valores por debajo del 70%.

Aplicación de la técnica de Normalización de Unidad al conjunto de datos de regresión muestra los siguientes resultados, ver Tabla 13.

Tabla 13: Aplicación de la Normalización de Unidad al Dataset de Regresión.

Dataset	Classifier	Normalización de Unidad			
		MSE	MAE	RMSE	R^2
02. Boston house price Dataset	1. Regresión Lineal	24.82	3.31	4.98	0.66
	2. Regresor Árbol de Decisión	11.85	2.64	3.44	0.84
	3. Random Forest	9.88	2.18	3.14	0.87
	4. Gradient Boosting	6.56	1.97	2.56	0.91
	5. SVM	37.18	3.61	6.1	0.49
	6. Redes Neuronales	25.27	3.13	5.03	0.66
	7. *k*NN	35.52	3.22	5.96	0.52
03. Body fat porcentaje estimates Dataset	1. Regresión Lineal	0.38	0.46	0.62	0.99
	2. Regresor Árbol de Decisión	0.41	0.34	0.64	0.99
	3. Random Forest	0.05	0.16	0.22	1
	4. Gradient Boosting	0.09	0.21	0.29	1
	5. SVM	10.23	2.1	3.2	0.78
	6. Redes Neuronales	21.67	3.13	4.66	0.53
	7. *k*NN	9.33	2.49	3.05	0.8

En base a la Tabla 13, los algoritmos que obtuvieron los resultados más destacados en términos de MSE en ambos conjuntos de datos son Random Forest y Gradient Boosting. En contraste, los restantes algoritmos como Regresión Lineal, Regresor de Árbol de Decisión, SVM y kNN exhibieron resultados mixtos. En algunos casos, el valor de MSE se acerca a 1, mientras que en otros casos se aleja significativamente del valor esperado.

*4.1.4. **Normalización Pareto Scaling***

Este método de normalización es usado para transformar variables numéricas a un rango en específico, la técnica calcula la media y la desviación estándar para luego ajustar cada valor mediante una formula especificada anteriormente. En la Tabla 14 se muestra los resultados de la aplicación de la técnica de normalización a los datasets de clasificación.

Tabla 14: Aplicación de la Normalización Pareto Scaling al Dataset de Clasificación.

Dataset	Classifier	Normalización Pareto Scaling					
		% Precisión	**Exactitud**	**Precisión**	**Sensibilidad**	**F1-score**	**AUC**
01. Madelon Dataset	1. Regresión Logística	52.69	0.53	0.52	0.56	0.54	0.53
	2. Árboles de decisión	76.54	0.77	0.74	0.80	0.77	0.77
	3. Random Forest	72.12	0.72	0.73	0.69	0.71	0.72
	4. Gradient Boosting	77.12	0.77	0.78	0.75	0.76	0.77

	5. SVM	55.58	0.56	0.55	0.54	0.55	0.56
	6. Redes Neuronales	52.69	0.53	0.54	0.28	0.37	0.52
	7. Naïve Bayes	59.62	0.60	0.59	0.58	0.58	0.60
	8. *k*NN	64.23	0.64	0.64	0.64	0.64	0.64
04. Heart Attack Análisis & Prediction Dataset	1. Regresión Logística	85.25	0.85	0.87	0.84	0.86	0.85
	2. Árboles de decisión	80.33	0.80	0.81	0.81	0.81	0.80
	3. Random Forest	83.61	0.84	0.87	0.81	0.84	0.84
	4. Gradient Boosting	78.69	0.79	0.81	0.78	0.79	0.79
	5. SVM	86.89	0.87	0.88	0.88	0.88	0.87
	6. Redes Neuronales	52.46	0.52	0.52	1.00	0.69	0.50
	7. Naïve Bayes	86.86	0.87	0.90	0.84	0.87	0.87
	8. *k*NN	73.77	0.74	0.75	0.75	0.75	0.74
05. Heart Failure Prediction Dataset	1. Regresión Logística	80.00	0.80	0.88	0.60	0.71	0.77
	2. Árboles de decisión	78.33	0.78	0.75	0.72	0.73	0.77
	3. Random Forest	73.33	0.73	0.76	0.52	0.62	0.70
	4. Gradient Boosting	73.33	0.73	0.76	0.52	0.62	0.70
	5. SVM	80.00	0.80	0.93	0.56	0.70	0.77
	6. Redes Neuronales	55.00	0.55	0.00	0.00	0.00	0.47
	7. Naïve Bayes	70.00	0.70	0.82	0.36	0.50	0.65
	8. *k*NN	70.00	0.70	0.73	0.44	0.55	0.66
06. Breast Cáncer Wisconsin Dataset	1. Regresión Logística	95.71	0.96	0.98	0.89	0.93	0.94
	2. Árboles de decisión	96.43	0.96	0.95	0.93	0.94	0.96
	3. Random Forest	96.43	0.96	0.95	0.93	0.94	0.96
	4. Gradient Boosting	95.71	0.96	0.95	0.91	0.93	0.95
	5. SVM	96.43	0.96	0.98	0.91	0.94	0.95
	6. Redes Neuronales	94.29	0.94	0.95	0.87	0.91	0.92
	7. Naïve Bayes	96.43	0.96	0.92	0.98	0.95	0.97
	8. *k*NN	97.14	0.97	0.98	0.93	0.95	0.96
07. Elevators Dataset	1. Regresión Logística	71.11	0.71	0.71	0.98	0.82	0.56
	2. Árboles de decisión	80.78	0.81	0.81	0.95	0.87	0.73
	3. Random Forest	84.79	0.85	0.85	0.95	0.90	0.79
	4. Gradient Boosting	84.88	0.85	0.84	0.96	0.90	0.78
	5. SVM	68.28	0.68	0.68	1.00	0.81	0.50
	6. Redes Neuronales	73.55	0.74	0.74	0.94	0.83	0.62
	7. Naïve Bayes	76.63	0.77	0.80	0.87	0.84	0.71
	8. *k*NN	67.14	0.67	0.71	0.86	0.78	0.56
08. Fried Dataset	1. Regresión Logística	83.91	0.84	0.84	0.84	0.84	0.84
	2. Árboles de decisión	82.14	0.82	0.85	0.78	0.81	0.82
	3. Random Forest	91.60	0.92	0.92	0.91	0.92	0.92
	4. Gradient Boosting	91.67	0.92	0.92	0.92	0.92	0.92
	5. SVM	84.14	0.84	0.84	0.84	0.84	0.84
	6. Redes Neuronales	92.64	0.93	0.95	0.90	0.92	0.93
	7. Naïve Bayes	86.77	0.87	0.86	0.88	0.87	0.87
	8. *k*NN	86.87	0.87	0.87	0.87	0.87	0.87
09. Puma32H Dataset	1. Regresión Logística	64.86	0.65	0.66	0.66	0.66	0.65
	2. Árboles de decisión	87.49	0.87	0.89	0.86	0.88	0.88
	3. Random Forest	87.49	0.87	0.89	0.87	0.88	0.88
	4. Gradient Boosting	87.43	0.87	0.89	0.87	0.88	0.87
	5. SVM	65.22	0.65	0.66	0.67	0.67	0.65
	6. Redes Neuronales	79.74	0.80	0.88	0.71	0.78	0.80
	7. Naïve Bayes	64.98	0.65	0.66	0.68	0.67	0.65
	8. *k*NN	60.04	0.60	0.62	0.61	0.61	0.60
10. Tumours of the central nervous system Dataset	1. Regresión Logística	66.67	0.67	0.00	0.00	0.00	0.50
	2. Árboles de decisión	66.67	0.67	0.50	0.50	0.50	0.62
	3. Random Forest	58.33	0.58	0.00	0.00	0.00	0.44
	4. Gradient Boosting	66.67	0.67	0.50	0.50	0.50	0.62
	5. SVM	66.67	0.67	0.00	0.00	0.00	0.50
	6. Redes Neuronales	33.33	0.33	0.33	1.00	0.50	0.50
	7. Naïve Bayes	75.00	0.75	0.60	0.75	0.67	0.75
	8. *k*NN	50.00	0.50	0.25	0.25	0.25	0.44

Los resultados de la normalización mediante Pareto Scaling, según se observa en la Tabla 14, indican que los algoritmos con un desempeño sobresaliente en general son Árboles de Decisión, Random Forest, Regresión Logística, Gradient Boosting y SVM. Estos algoritmos alcanzan valores de precisión (Accuracy) por encima del 80%. De manera similar, un algoritmo mostró resultados mixtos, es decir, mantuvo valores por encima del 80% en algunos conjuntos de datos y por debajo del 70% en otros. Por otro lado, los algoritmos que mayormente obtuvieron resultados por debajo del 70% en la mayoría de los conjuntos de datos fueron Redes Neuronales y *k*NN.

Aplicación de la técnica de normalización de Unidad al conjunto de datos de regresión Tabla 15.

Tabla 15: Aplicación de la Normalización de Pareto Scaling al Dataset de Regresión.

Dataset	Classifier	Pareto Scaling			
		MSE	**MAE**	**RMSE**	R^2
02. Boston house price Dataset	1. Regresión Lineal	24.82	3.31	4.98	0.66
	2. Regresor Árbol de Decisión	11.11	2.55	3.33	0.85
	3. Random Forest	9.27	2.14	3.04	0.87
	4. Gradient Boosting	6.63	1.98	2.57	0.91
	5. SVM	42.61	3.78	6.53	0.42
	6. Redes Neuronales	23.59	3.47	4.86	0.68
	7. *k*NN	27.77	3.01	5.27	0.62
03. Body fat porcentaje estimates Dataset	1. Regresión Lineal	0.38	0.46	0.62	0.99
	2. Regresor Árbol de Decisión	0.43	0.35	0.65	0.99
	3. Random Forest	0.05	0.16	0.21	1.00
	4. Gradient Boosting	0.08	0.21	0.28	1.00
	5. SVM	27.45	4.46	5.24	0.41
	6. Redes Neuronales	26.90	4.42	5.19	0.42
	7. *k*NN	23.24	3.93	4.82	0.50

En ambos datasets según la Tabla 15, los algoritmos exhibieron resultados variados; sin embargo, aquellos que estuvieron más próximos al valor de referencia del MSE 1 fueron el Regresor de Árbol de Decisión, Random Forest y Gradient Boosting. Por otra parte, los algoritmos que mostraron valores significativamente más alejados del objetivo fueron Regresión Lineal, SVM, Redes Neuronales y *k*NN, con valores que incluso llegaron a alcanzar un MSE de 42.61 en el caso de SVM.

*4.1.5. **Normalización Sigmoidea***

Esta técnica de normalización sigmoidea transforma los valores de una variable a un rango específico y limitado, mapea los valores en un rango de 0 y 1 utilizando la función sigmoide. Al aplicar esta técnica los valores más pequeños se acercan a 0 y los valores más grandes se acercan a 1, ver la Tabla 16.

Tabla 16: Aplicación de la Normalización Sigmoidea al Dataset de Clasificación.

Dataset	Classifier	Normalización Sigmoidea					
		% Precisión	Exactitud	Precisión	Sensibilidad	F1-score	AUC
01. Madelon Dataset	1. Regresión Logística	49.42	0.49	0.49	1.00	0.66	0.50
	2. Árboles de decisión	49.42	0.49	0.49	1.00	0.66	0.50
	3. Random Forest	49.42	0.49	0.49	1.00	0.66	0.50
	4. Gradient Boosting	49.42	0.49	0.49	1.00	0.66	0.50
	5. SVM	49.42	0.49	0.49	1.00	0.66	0.50
	6. Redes Neuronales	49.42	0.49	0.49	1.00	0.66	0.50
	7. Naïve Bayes	49.42	0.49	0.49	1.00	0.66	0.50
	8. *k*NN	50.58	0.49	0.49	1.00	0.66	0.50
04. Heart Attack Análisis & Prediction Dataset	1. Regresión Logística	91.80	0.92	0.91	0.94	0.92	0.92
	2. Árboles de decisión	77.05	0.77	0.80	0.75	0.77	0.77
	3. Random Forest	81.97	0.82	0.84	0.81	0.83	0.82
	4. Gradient Boosting	81.97	0.82	0.84	0.81	0.83	0.82
	5. SVM	90.16	0.90	0.88	0.94	0.91	0.90
	6. Redes Neuronales	83.61	0.84	0.87	0.81	0.84	0.84
	7. Naïve Bayes	81.97	0.82	0.86	0.78	0.82	0.82
	8. *k*NN	81.97	0.82	0.86	0.78	0.82	0.82
05. Heart Failure Prediction Dataset	1. Regresión Logística	58.33	0.58	0.00	0.00	0.00	0.50
	2. Árboles de decisión	68.33	0.68	0.88	0.28	0.42	0.63
	3. Random Forest	73.33	0.73	0.76	0.52	0.62	0.70
	4. Gradient Boosting	68.33	0.68	0.71	0.40	0.51	0.64
	5. SVM	58.33	0.58	0.00	0.00	0.00	0.50
	6. Redes Neuronales	58.33	0.58	0.00	0.00	0.00	0.50
	7. Naïve Bayes	63.33	0.63	1.00	0.12	0.21	0.56
	8. *k*NN	58.33	0.58	0.50	0.16	0.24	0.52
06. Breast Cáncer Wisconsin Dataset	1. Regresión Logística	98.57	0.99	0.96	1	0.98	0.99
	2. Árboles de decisión	96.43	0.96	0.95	0.93	0.94	0.96
	3. Random Forest	97.86	0.98	0.98	0.96	0.97	0.97
	4. Gradient Boosting	95.71	0.96	0.95	0.91	0.93	0.95
	5. SVM	98.57	0.99	0.96	1	0.98	0.99
	6. Redes Neuronales	97.14	0.97	0.94	0.98	0.96	0.97
	7. Naïve Bayes	98.57	0.99	0.96	1	0.98	0.99
	8. *k*NN	97.86	0.98	0.96	0.98	0.97	0.98
07. Elevators Dataset	1. Regresión Logística	71.08	0.71	0.71	0.98	0.82	0.56
	2. Árboles de decisión	80.78	0.81	0.81	0.95	0.87	0.73
	3. Random Forest	85.24	0.85	0.85	0.95	0.90	0.80
	4. Gradient Boosting	84.88	0.85	0.84	0.96	0.90	0.78
	5. SVM	68.28	0.68	0.68	1.00	0.81	0.50
	6. Redes Neuronales	73.55	0.74	0.74	0.94	0.83	0.62
	7. Naïve Bayes	76.63	0.77	0.80	0.87	0.84	0.71
	8. *k*NN	67.14	0.67	0.71	0.86	0.78	0.56
08. Fried Dataset	1. Regresión Logística	84.16	0.84	0.84	0.84	0.84	0.84
	2. Árboles de decisión	82.14	0.82	0.85	0.78	0.81	0.82
	3. Random Forest	91.70	0.92	0.92	0.91	0.92	0.92
	4. Gradient Boosting	91.67	0.92	0.91	0.92	0.92	0.92
	5. SVM	84.41	0.84	0.85	0.84	0.84	0.84

	6. Redes Neuronales	86.83	0.87	0.89	0.84	0.86	0.87
	7. Naïve Bayes	87.14	0.87	0.86	0.89	0.87	0.87
	8. *k*NN	86.79	0.87	0.87	0.87	0.87	0.87
09. Puma32H Dataset	1. Regresión Logística	64.92	0.65	0.66	0.66	0.66	0.65
	2. Árboles de decisión	87.00	0.87	0.89	0.86	0.87	0.87
	3. Random Forest	87.25	0.87	0.89	0.86	0.88	0.87
	4. Gradient Boosting	87.80	0.88	0.89	0.88	0.88	0.88
	5. SVM	64.98	0.65	0.66	0.66	0.66	0.65
	6. Redes Neuronales	81.45	0.81	0.76	0.93	0.84	0.81
	7. Naïve Bayes	64.92	0.65	0.66	0.66	0.66	0.65
	8. *k*NN	63.94	0.64	0.66	0.64	0.65	0.64
10. Tumours of the central nervous system Dataset	1. Regresión Logística	66.67	0.67	0.00	0.00	0.00	0.50
	2. Árboles de decisión	75.00	0.75	0.67	0.50	0.57	0.69
	3. Random Forest	66.67	0.67	0.00	0.00	0.00	0.50
	4. Gradient Boosting	66.67	0.67	0.50	0.50	0.50	0.62
	5. SVM	66.67	0.67	0.00	0.00	0.00	0.50
	6. Redes Neuronales	66.67	0.67	0.00	0.00	0.00	0.50
	7. Naïve Bayes	66.67	0.67	0.00	0.00	0.00	0.50
	8. *k*NN	58.33	0.58	0.00	0.00	0.00	0.44

Se aplicaron los 8 algoritmos de aprendizaje automático al conjunto de datos normalizado mediante la técnica Sigmoidea (Tabla 16). Entre estos algoritmos, Regresión Logística, Árboles de Decisión, Random Forest, Gradient Boosting y Redes Neuronales exhibieron valores de precisión (Accuracy) superiores al 80%. Por otro lado, Naïve Bayes, *k*NN, SVM y Regresión Logística obtuvieron resultados por debajo del 70%, mostrando igualmente rendimientos mixtos.

Aplicación de la técnica Sigmoidea a los conjuntos de datos de regresión y los valores se muestran a continuación, ver Tabla 17.

Tabla 17: Aplicación de la Normalización Sigmoidea al Dataset de Regresión.

Dataset	**Classifier**	**Normalización Sigmoidea**			
		MSE	**MAE**	**RMSE**	R^2
02. Boston house price Dataset	1. Regresión Lineal	40.94	4.00	6.40	0.44
	2. Regresor Árbol de Decisión	10.65	2.53	3.26	0.85
	3. Random Forest	7.41	1.94	2.72	0.90
	4. Gradient Boosting	9.17	2.10	3.03	0.88
	5. SVM	55.99	4.74	7.48	0.24
	6. Redes Neuronales	79.70	6.62	8.93	-0.09
	7. *k*NN	41.03	3.51	6.41	0.44
03. Body fat porcentaje estimates Dataset	1. Regresión Lineal	0.09	0.26	0.30	1.00
	2. Regresor Árbol de Decisión	0.36	0.25	0.60	0.99
	3. Random Forest	0.10	0.17	0.32	1.00
	4. Gradient Boosting	0.32	0.25	0.56	0.99

	5. SVM	48.14	5.63	6.94	-0.03
	6. Redes Neuronales	46.66	5.56	6.83	0.00
	7. *k*NN	0.11	0.21	0.33	1.00

De acuerdo a la Tabla 17 de los algoritmos aplicados al conjunto de datos de Regresión, se observan resultados diversos. En el caso de Regresión Logística, se obtienen valores de MSE de 40.94 y 0.09, mientras que Árboles de Decisión arroja 10.65 y 0.36. En cuanto a Random Forest y Gradient Boosting, los resultados fluctúan en el rango de 0.10 a 9.17. Por otro lado, los algoritmos SVM, Redes Neuronales y *k*NN exhiben variabilidad en sus resultados en función del conjunto de datos.

En resumen, la elección de la técnica de normalización depende del problema, los requisitos del algoritmo de Machine Learning y la naturaleza de los datos. La normalización de datos permite que los algoritmos de Machine Learning funcionen de manera más eficiente y precisa al reducir los efectos de las diferencias de escala y distribución entre las características. Ayuda a evitar que características con valores más grandes dominen aquellas con valores más pequeños y facilita la convergencia del modelo durante el entrenamiento.

5. RESULTADOS

En esta sección se presentan los hallazgos obtenidos al aplicar las 5 técnicas de normalización a los 10 conjuntos de datos y la aplicación de los algoritmos de aprendizaje automático (ML). Los resultados se muestran en tablas y gráficas de barras apiladas, se analiza el impacto de las diferentes técnicas de normalización en el rendimiento de los algoritmos de ML.

5.1. Análisis de Resultados del Datasets Antes de la Normalización

En la siguiente Tabla 18 se muestran los valores de las métricas de evaluación de los algoritmos antes de aplicar las técnicas de normalización a los conjuntos de datos.

Tabla 18: Resultado Dataset de Clasificación Antes de la Normalización.

Dataset	Classifier	Sin Aplicar la Normalización					
		% Precisión	Exactitud	Precisión	Sensibilidad	F1-score	AUC
01. Madelon Dataset	1. Regresión Logística	52.31	0.52	0.52	0.55	0.53	0.52
	2. Árboles de decisión	76.35	0.76	0.74	0.80	0.77	0.76
	3. Random Forest	71.92	0.72	0.72	0.70	0.71	0.72
	4. Gradient Boosting	77.31	0.77	0.78	0.75	0.77	0.77
	5. SVM	55.57	0.78	0.79	0.71	0.73	0.78
	6. Redes Neuronales	49.42	0.49	0.49	1.00	0.66	0.50
	7. Naïve Bayes	59.62	0.60	0.59	0.58	0.58	0.60
	8. *k*NN	69.62	0.70	0.70	0.68	0.69	0.70
04. Heart Attack Análisis & Prediction Dataset	1. Regresión Logística	88.52	0.89	0.88	0.91	0.89	0.88
	2. Árboles de decisión	83.61	0.84	0.84	0.84	0.84	0.84
	3. Random Forest	86.89	0.87	0.85	0.91	0.88	0.87
	4. Gradient Boosting	77.05	0.77	0.80	0.75	0.77	0.77
	5. SVM	86.89	0.87	0.88	0.88	0.88	0.87
	6. Redes Neuronales	52.46	0.52	0.52	1.00	0.69	0.50
	7. Naïve Bayes	86.89	0.87	0.90	0.84	0.87	0.87
	8. *k*NN	65.85	0.69	0.69	0.75	0.72	0.69
05. Heart Failure Prediction Dataset	1. Regresión Logística	80.00	0.80	0.88	0.88	0.71	0.77
	2. Árboles de decisión	78.33	0.78	0.75	0.72	0.73	0.77
	3. Random Forest	75.00	0.75	0.81	0.52	0.63	0.72
	4. Gradient Boosting	73.33	0.73	0.76	0.52	0.62	0.70
	5. SVM	75.00	0.75	0.81	0.52	0.63	0.72
	6. Redes Neuronales	41.67	0.42	0.42	1.00	0.59	0.50
	7. Naïve Bayes	73.33	0.73	0.91	0.40	0.56	0.69
	8. *k*NN	53.33	0.53	0.29	0.08	0.12	0.47
06. Breast Cáncer Wisconsin Dataset	1. Regresión Logística	97.14	0.97	0.98	0.93	0.95	0.96
	2. Árboles de decisión	96.43	0.96	0.95	0.93	0.94	0.96
	3. Random Forest	95.71	0.96	0.98	0.89	0.93	0.94
	4. Gradient Boosting	95.71	0.96	0.95	0.91	0.93	0.95
	5. SVM	96.43	0.96	0.98	0.91	0.94	0.95
	6. Redes Neuronales	93.57	0.94	0.95	0.84	0.89	0.91
	7. Naïve Bayes	96.43	0.96	0.92	0.98	0.95	0.97
	8. *k*NN	98.57	0.99	0.98	0.98	0.98	0.98

07. Elevators Dataset	1. Regresión Logística	75.87	0.76	0.77	0.92	0.84	0.67
	2. Árboles de decisión	80.30	0.80	0.80	0.95	0.87	0.72
	3. Random Forest	85.75	0.86	0.85	0.95	0.90	0.80
	4. Gradient Boosting	84.70	0.85	0.84	0.96	0.90	0.78
	5. SVM	83.80	0.81	0.82	0.85	0.82	0.80
	6. Redes Neuronales	74.73	0.75	0.77	0.90	0.83	0.66
	7. Naïve Bayes	73.64	0.74	0.77	0.88	0.82	0.66
	8. *k*NN	68.92	0.69	0.73	0.87	0.79	0.59
08. Fried Dataset	1. Regresión Logística	83.89	0.84	0.84	0.84	0.84	0.84
	2. Árboles de decisión	82.14	0.82	0.85	0.78	0.81	0.82
	3. Random Forest	91.64	0.92	0.92	0.92	0.92	0.92
	4. Gradient Boosting	91.67	0.92	0.92	0.92	0.92	0.92
	5. SVM	84.17	0.84	0.84	0.84	0.84	0.84
	6. Redes Neuronales	88.30	0.88	0.88	0.89	0.88	0.88
	7. Naïve Bayes	86.77	0.87	0.86	0.88	0.87	0.87
	8. *k*NN	86.60	0.87	0.86	0.87	0.87	0.87
09. Puma32H Dataset	1. Regresión Logística	64.86	0.65	0.66	0.66	0.66	0.65
	2. Árboles de decisión	87.49	0.87	0.89	0.86	0.88	0.88
	3. Random Forest	87.49	0.87	0.89	0.86	0.88	0.88
	4. Gradient Boosting	87.43	0.87	0.89	0.87	0.88	0.87
	5. SVM	85.44	0.85	0.85	0.84	0.88	0.84
	6. Redes Neuronales	61.01	0.61	0.61	0.67	0.64	0.61
	7. Naïve Bayes	64.98	0.65	0.66	0.68	0.67	0.65
	8. *k*NN	58.15	0.58	0.60	0.57	0.59	0.58
10. Tumours of the central nervous system Dataset	1. Regresión Logística	83.33	0.83	1.00	0.50	0.67	0.75
	2. Árboles de decisión	58.33	0.58	0.43	0.75	0.55	0.62
	3. Random Forest	66.67	0.67	0.50	0.25	0.33	0.56
	4. Gradient Boosting	66.67	0.67	0.50	0.50	0.50	0.62
	5. SVM	83.33	0.83	1.00	0.50	0.67	0.75
	6. Redes Neuronales	33.33	0.33	0.33	1.00	0.50	0.50
	7. Naïve Bayes	75.00	0.75	0.60	0.75	0.67	0.75
	8. *k*NN	41.67	0.42	0.00	0.00	0.00	0.31

Antes de la normalización, el conjunto de datos de clasificación refleja que los algoritmos de Regresión Logística, Árboles de Decisión, Random Forest, Gradient Boosting y SVM superan el 80% en precisión (Accuracy). En contraste, el Naïve Bayes arroja resultados variados: en 3 conjuntos de datos, los valores superan el 80%, en otros 3 se encuentran en el rango del 70% al 80%, mientras que, en 2 conjuntos de datos, los valores caen por debajo del 70%. Asimismo, tanto el algoritmo de Redes Neuronales como el de *k*NN muestran una precisión mayoritariamente inferior al 70% en la mayoría de los conjuntos de datos.

Tabla 19: Resultado del Dataset Regresión Antes de la Normalización

Dataset	Classifier	Sin Aplicar la normalización			
		MSE	**MAE**	**RMSE**	R^2
02. Boston house price Dataset	1. Regresión Lineal	24.82	3.31	4.98	0.66
	2. Regresor Árbol de Decisión	10.49	2.44	3.24	0.86
	3. Random Forest	8.62	2.03	2.94	0.88
	4. Gradient Boosting	6.48	1.96	2.55	0.91
	5. SVM	52.93	4.54	7.28	0.28
	6. Redes Neuronales	34.42	4.03	5.87	0.53
	7. *k*NN	26.21	3.70	5.12	0.64
03. Body fat porcentaje estimates Dataset	1. Regresión Lineal	0.38	0.46	0.62	0.99
	2. Regresor Árbol de Decisión	2.00	0.57	1.42	0.96
	3. Random Forest	0.05	0.15	0.21	1.00
	4. Gradient Boosting	0.08	0.21	0.29	1.00
	5. SVM	30.35	4.64	5.51	0.35
	6. Redes Neuronales	27.32	4.38	5.23	0.41
	7. *k*NN	24.31	3.94	4.93	0.48

De manera similar, en la Tabla 19 se aprecia una diversidad de resultados entre los algoritmos aplicados al conjunto de datos de regresión en su version original. Por ejemplo, algoritmos como la Regresión Lineal, Random Forest y Gradient Boosting exhiben valores de MSE cercanos a 1 en algunos conjuntos de datos, mientras que en otros muestran desviaciones significativas con respecto al valor objetivo. Además, se observa que los algoritmos SVM, Redes Neuronales y *k*NN presentan valores de MSE considerablemente alejados de aproximarse a 1, para una mayor claridad ver la gráfica. Las razones por las cuales los valores no cumplen con las expectativas podrían incluir la necesidad de aumentar el entrenamiento de estos algoritmos o mejorar los recursos computacionales disponibles, entre otros factores.

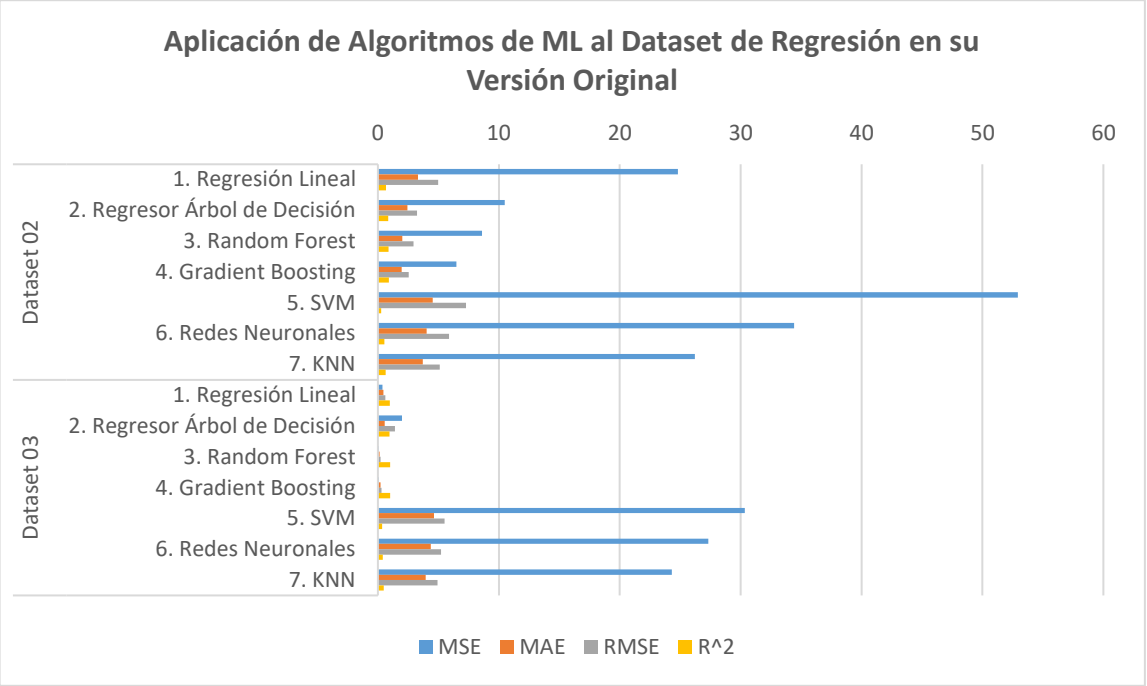


Figura 2: Resultado del Dataset Regresión Antes de la Normalización

Tal como se mencionó anteriormente en la gráfica se puede ver que los algoritmos con valores MSE mucho más bajos en ambos conjuntos de datos son Regresión Lineal, Regresor Árbol de decisión, Random Forest y Gradient Boosting, esto indica que son los algoritmos con mejores resultados en precisión.

En las siguientes gráficas apiladas del conjunto de datos de clasificación en su versión original (Figura 1 – Parte 1 y Parte 2) se observa a los algoritmos con mejores resultados antes de la normalización. Cabe mencionar que los nombres de los dataset fueron remplazados por su Id según la Tabla 4 página 24.

Aplicación de Algoritmos de ML al Dataset Original

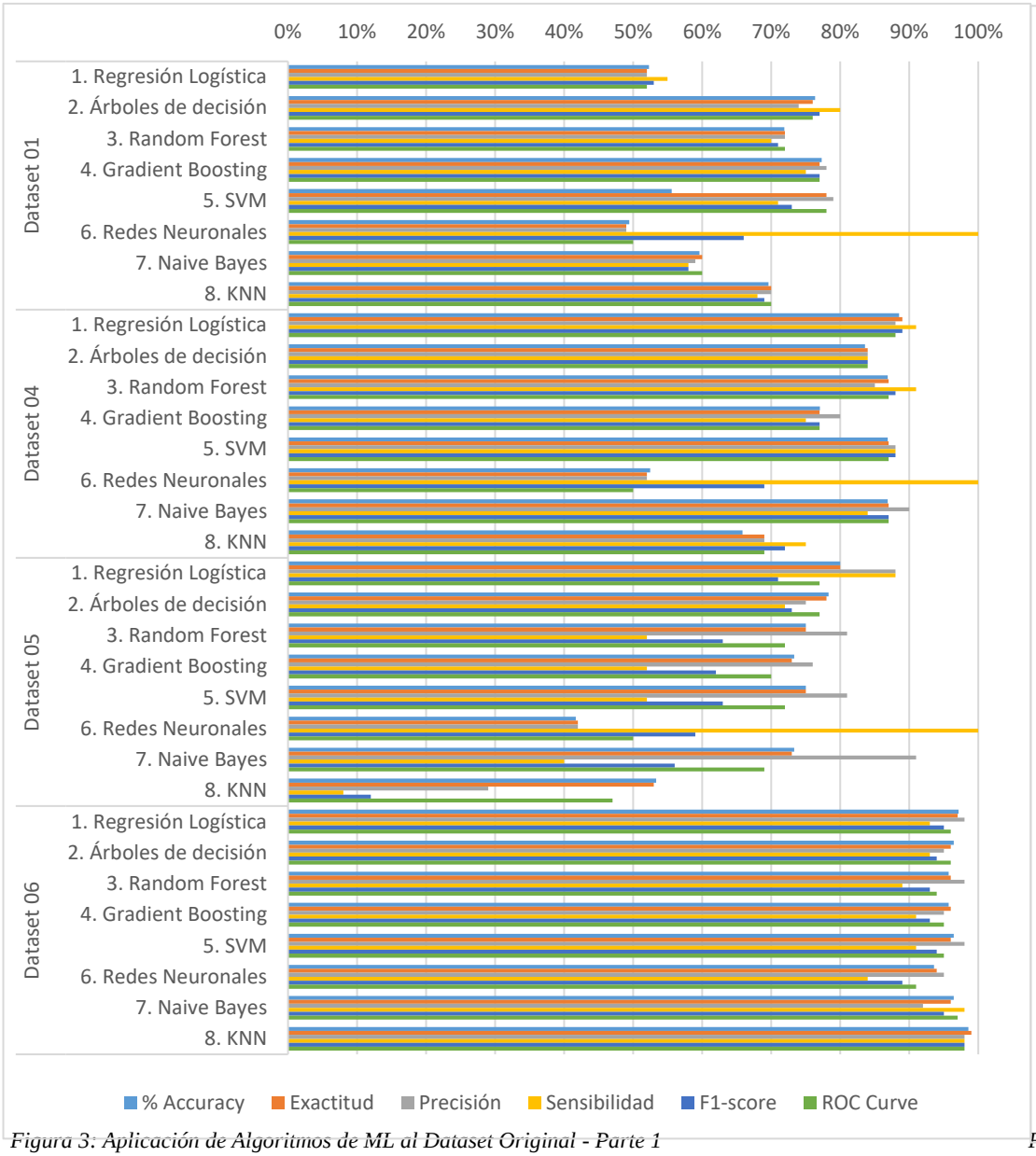


Figura 3: Aplicación de Algoritmos de ML al Dataset Original - Parte 1

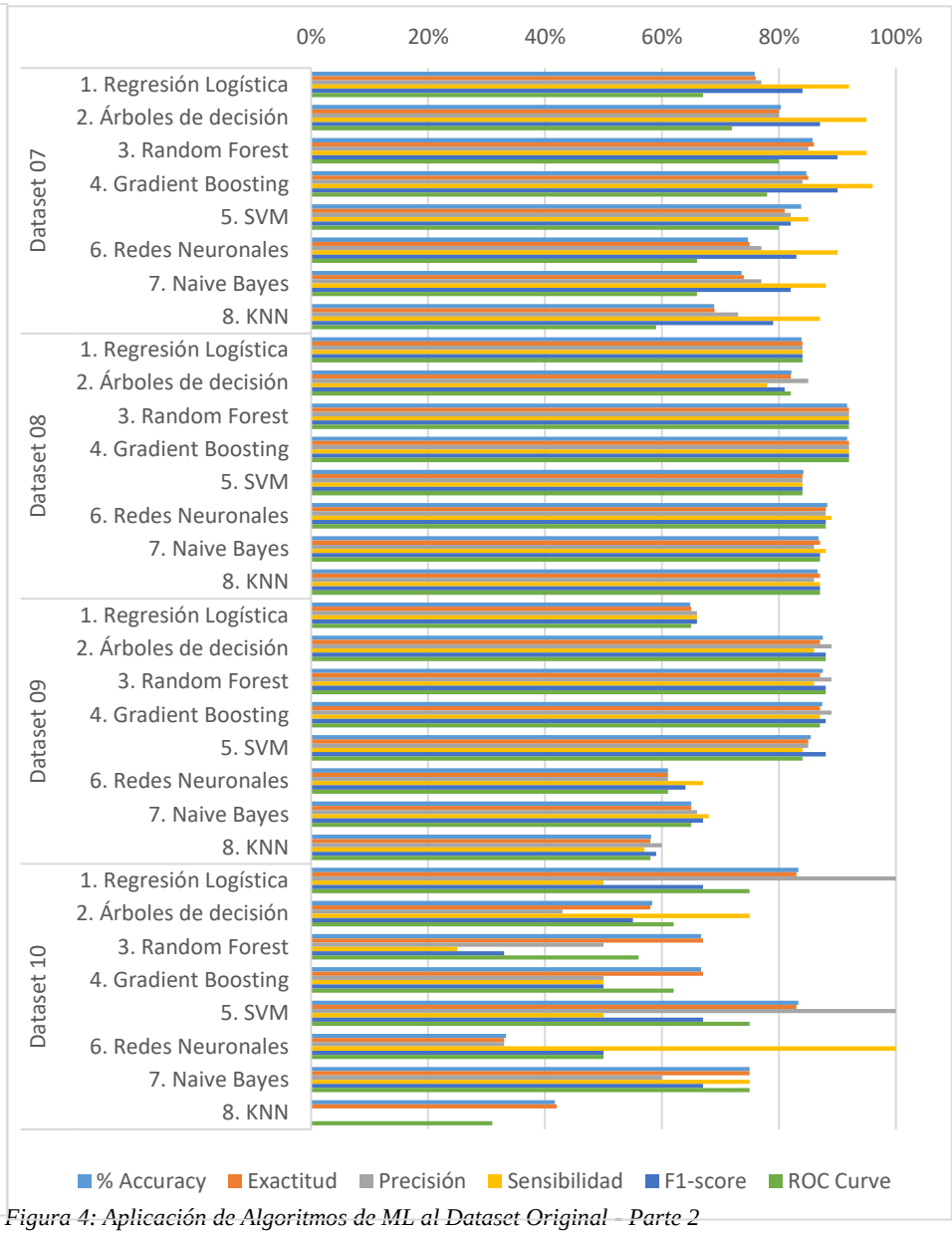


Figura 4: Aplicación de Algoritmos de ML al Dataset Original - Parte 2

Las gráficas (Figura 2 – Parte 1 y Parte 3) revelan que los algoritmos con los mejores resultados en las métricas de precisión, exactitud, sensibilidad, F1-Score y curva ROC, superan el umbral del 80% en el conjunto de datos N°.04. Estos algoritmos incluyen Regresión Logística, Árboles de Decisión, Random Forest, SVM y Naïve Bayes. En el conjunto de datos N°.06, se destacan Regresión Logística, Árboles de Decisión, Random Forest, Gradient Boosting, SVM, Redes Neuronales, Naïve Bayes y *k*NN. Similarmente, en el conjunto de datos N°.08, los algoritmos notables son Regresión Logística, Random Forest, Gradient Boosting, SVM, Redes Neuronales, Naïve Bayes y *k*NN. Por otro lado, en el conjunto de datos N°.09, se observan Árboles de Decisión, Random Forest, Gradient Boosting y SVM como los más destacados. No obstante, en otros conjuntos de datos, las métricas registran resultados inferiores al 80%. Estos resultados reflejan el desempeño de los algoritmos previo a la aplicación de las técnicas de normalización. Se anticipa que después de la normalización, se mantengan o superen los valores por encima del 80%.

A continuación, se detallan las gráficas apiladas posteriores a la implementación de diversas técnicas de normalización.

5.2. Análisis de Resultados del Datasets Normalizados

Se aplicaron 5 técnicas de normalización a un total de 10 conjuntos de datos (8 conjuntos de datos para clasificación y 2 conjuntos de datos para regresión). Posteriormente, se emplearon 8 algoritmos de aprendizaje automático para realizar análisis.

Para los conjuntos de datos de clasificación los resultados obtenidos se agruparon en 3 categorías: en la primera, se ubicaron los algoritmos cuyas métricas superaron el umbral del 80%; en la segunda, aquellos que mantuvieron sus métricas dentro del rango del 60% al 79%; por último, en la tercera, se incluyeron los algoritmos cuyas métricas de evaluación se situaron por debajo del 59%.

A continuación, se presentan las gráficas correspondientes a cada una de estas categorías.

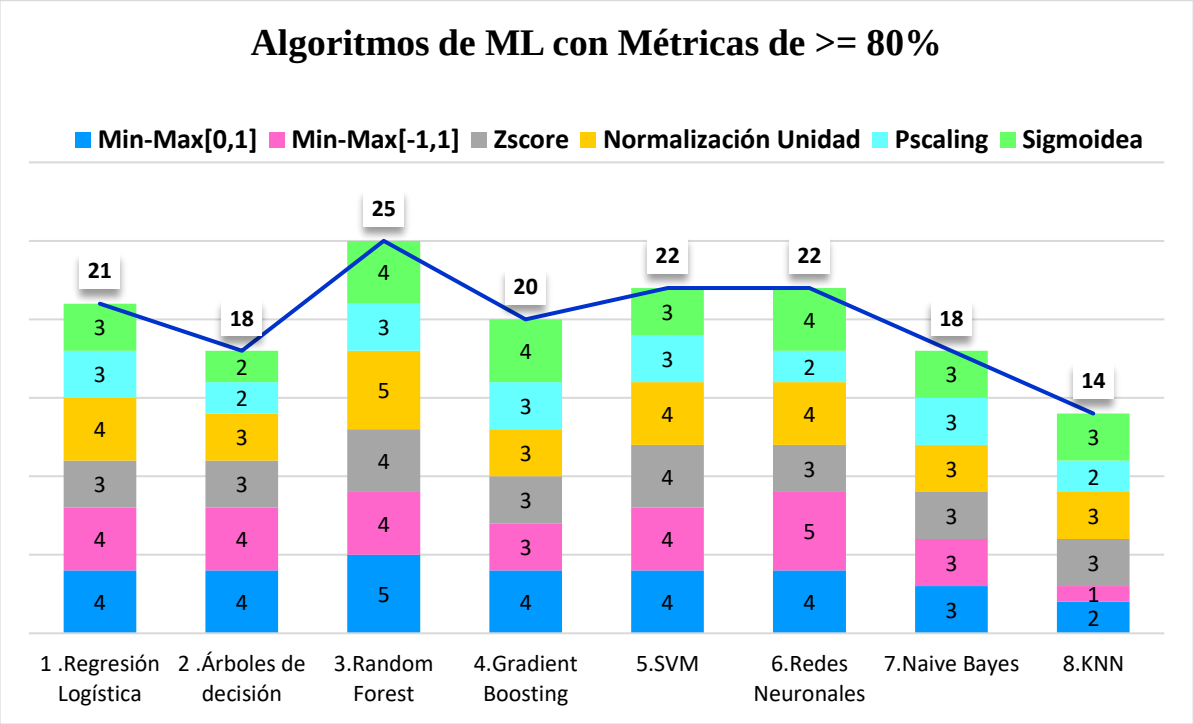


Figura 5: Algoritmos de ML con Métricas >= 80%

De acuerdo al análisis de la gráfica ilustrada (Figura 4) resultados obtenidos por diversos algoritmos cuyas métricas superan el umbral del 80%, surge un patrón intrigante. El algoritmo Random Forest se destaca por haber superado este umbral en 25 ocasiones, consolidándose como el líder preeminente en términos de frecuencia de éxito. A continuación, encontramos a SVM y a las Redes Neuronales, ambas con 22 ocasiones en las que han logrado exceder este valor de referencia. Por otro lado, la Regresión Logística ha demostrado su desempeño en 21 ocasiones, mientras que tanto Naïve Bayes como Árboles de Decisión han conseguido rebasar el umbral del 80% en 18 ocasiones. En contraste, *k*NN solo ha logrado cruzar esta línea en 14 ocasiones, lo que sugiere que es el algoritmo que menos veces ha alcanzado este nivel requerido de rendimiento.

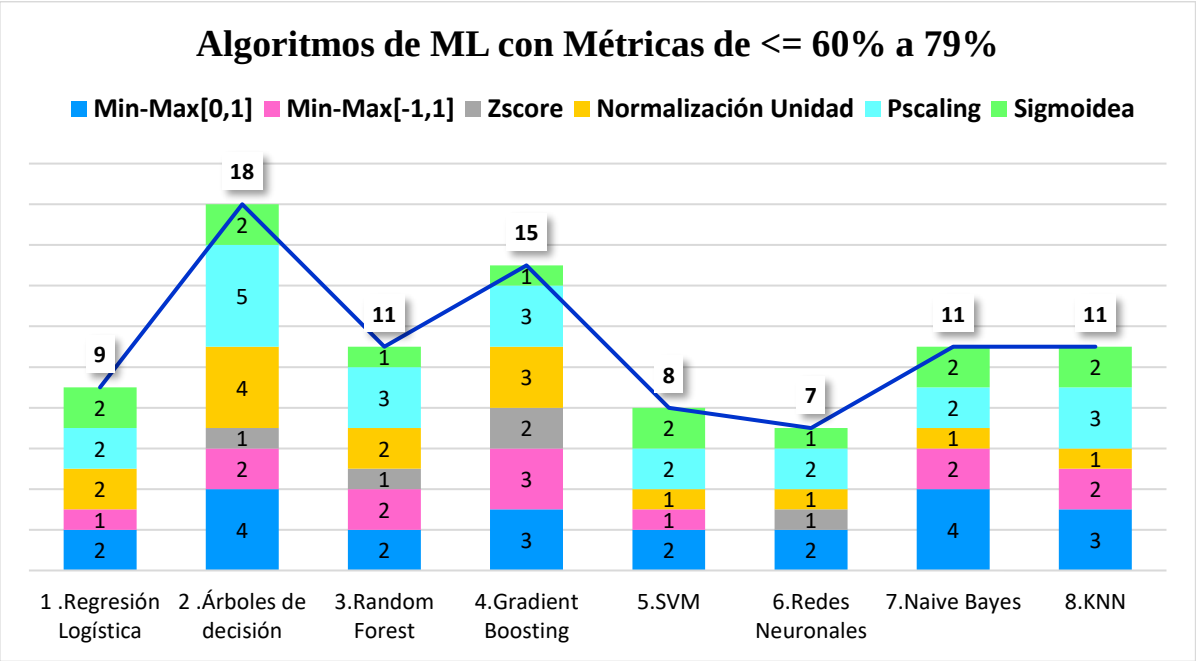


Figura 6: Algoritmos de ML con Métricas <= 60% a 79%

Al analizar la gráfica (Figura 5) de los algoritmos que sus métricas se encuentran en el rango del 60% al 79%, se presentan algunas tendencias notables. Específicamente, se destaca que el algoritmo de Árboles de Decisión ha superado el umbral del 60% en 18 ocasiones, considerándose como el líder en este grupo de 8 algoritmos. En un segundo lugar, se encuentra el algoritmo de Gradient Boosting con 15 ocasiones en las que ha logrado cumplir con esta condición. Posteriormente, se observa un trío conformado por los algoritmos Random Forest, Naïve Bayes y *k*NN, todos ellos superando el umbral en 11 ocasiones. Con una presencia ligeramente más moderada, los algoritmos de Regresión Logística lograron alcanzar este estándar en 9 ocasiones, mientras que SVM lo hizo en 8, y las Redes Neuronales en 7. Es importante señalar que estos tres últimos algoritmos presentan una sutil variabilidad en cuanto a la frecuencia de cumplimiento.

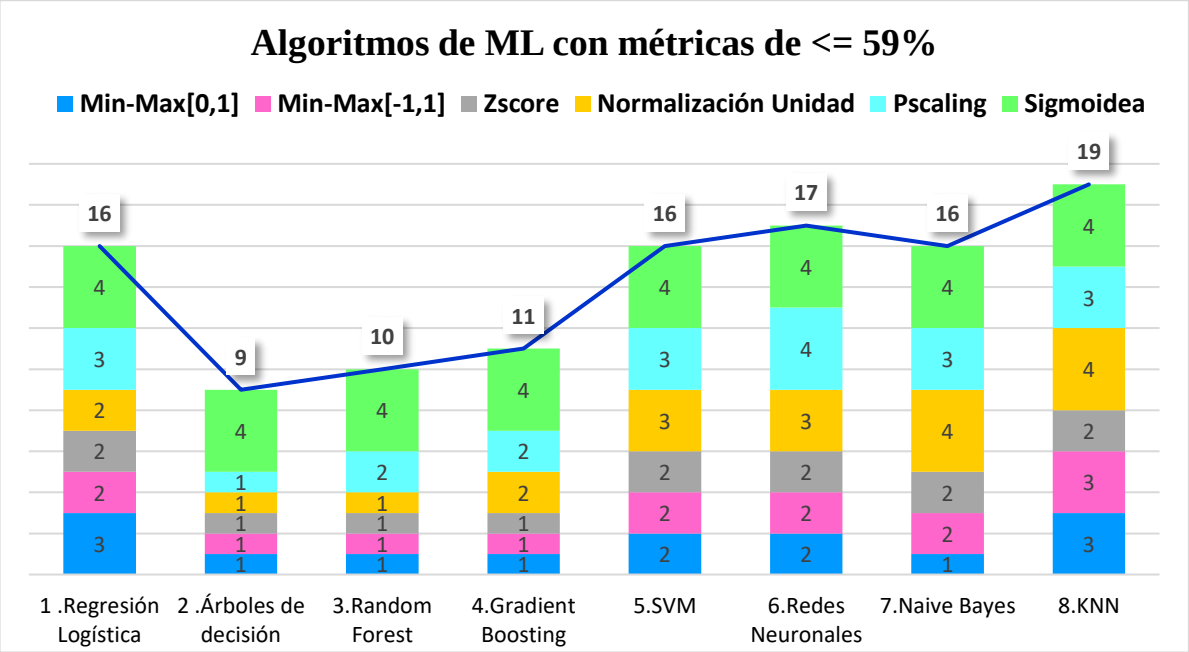


Figura 7: Algoritmos de ML con métricas <= 59%

Basado en el análisis de la gráfica (Figura 6) sobre el desempeño de los algoritmos con métricas que caen por debajo del 59%. En este contexto, se destaca a *k*NN como líder en este aspecto, habiendo demostrado un rendimiento por debajo del umbral en 19 ocasiones. Le sigue en este ranking Redes Neuronales, con 17 instancias en las que su rendimiento se mantuvo por debajo del valor establecido. De manera similar, Regresión Logística, SVM y Naïve Bayes comparten la marca de 16 ocasiones en las que sus métricas no lograron alcanzar el umbral. Por otro lado, Gradient Boosting exhibió este comportamiento en 11 ocasiones, mientras que Random Forest lo hizo en 10 ocasiones, y Árboles de Decisión en 9 ocasiones. Esto implica que, en términos de frecuencia, *k*NN es el algoritmo que ha registrado más instancias de desempeño por debajo del umbral en comparación con los demás algoritmos analizados.

A continuación, se observa a mayor detalle el comportamiento de los algoritmos por cada técnica de normalización y conjuntos de datos de clasificación utilizados para esta investigación.

Normalización de Mín-Máx [0,1] & Aplicación de Algoritmos de ML

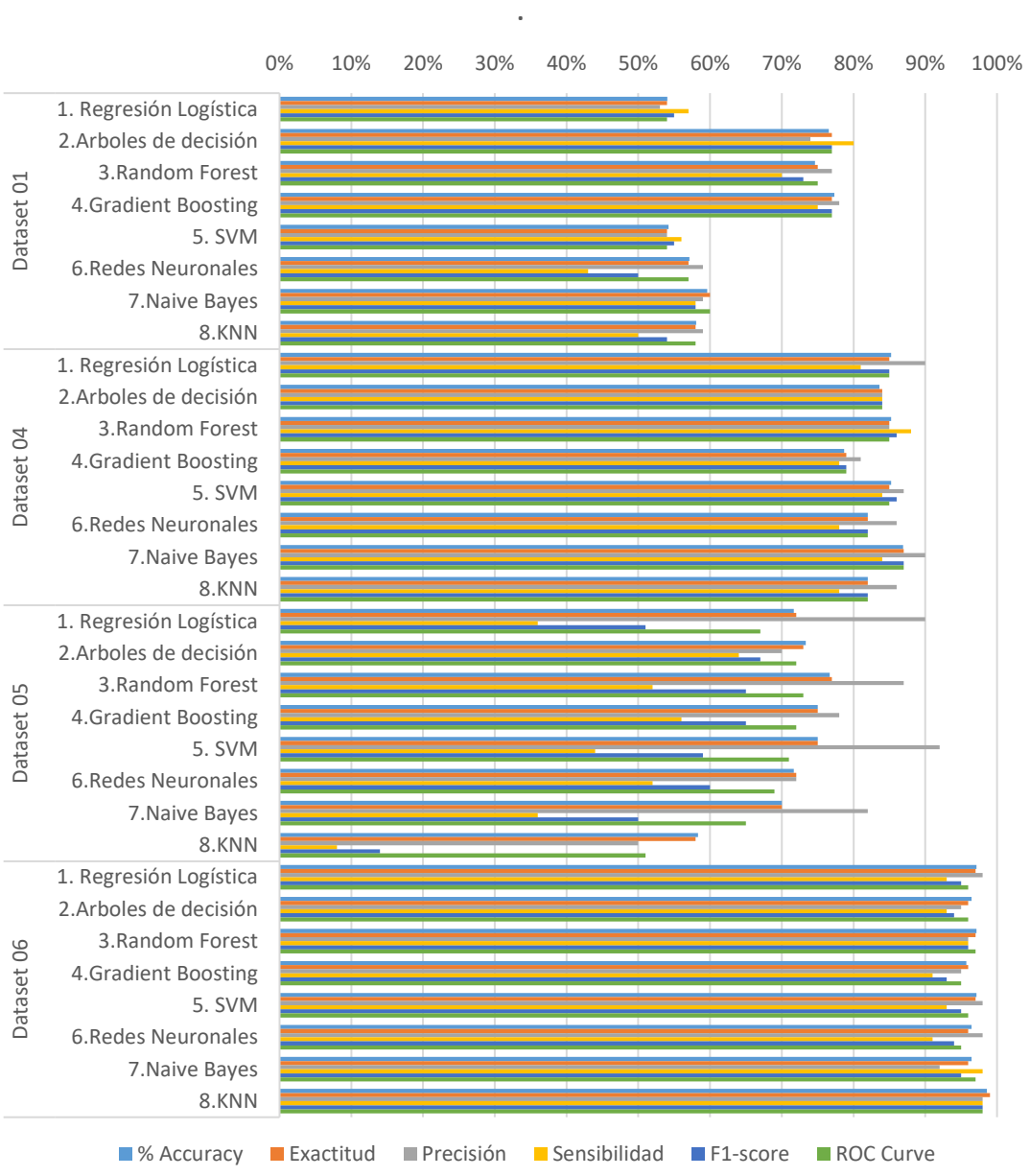


Figura 9: Normalización de Mín-Máx [0,1] & Aplicación de Algoritmo de ML- Parte 1

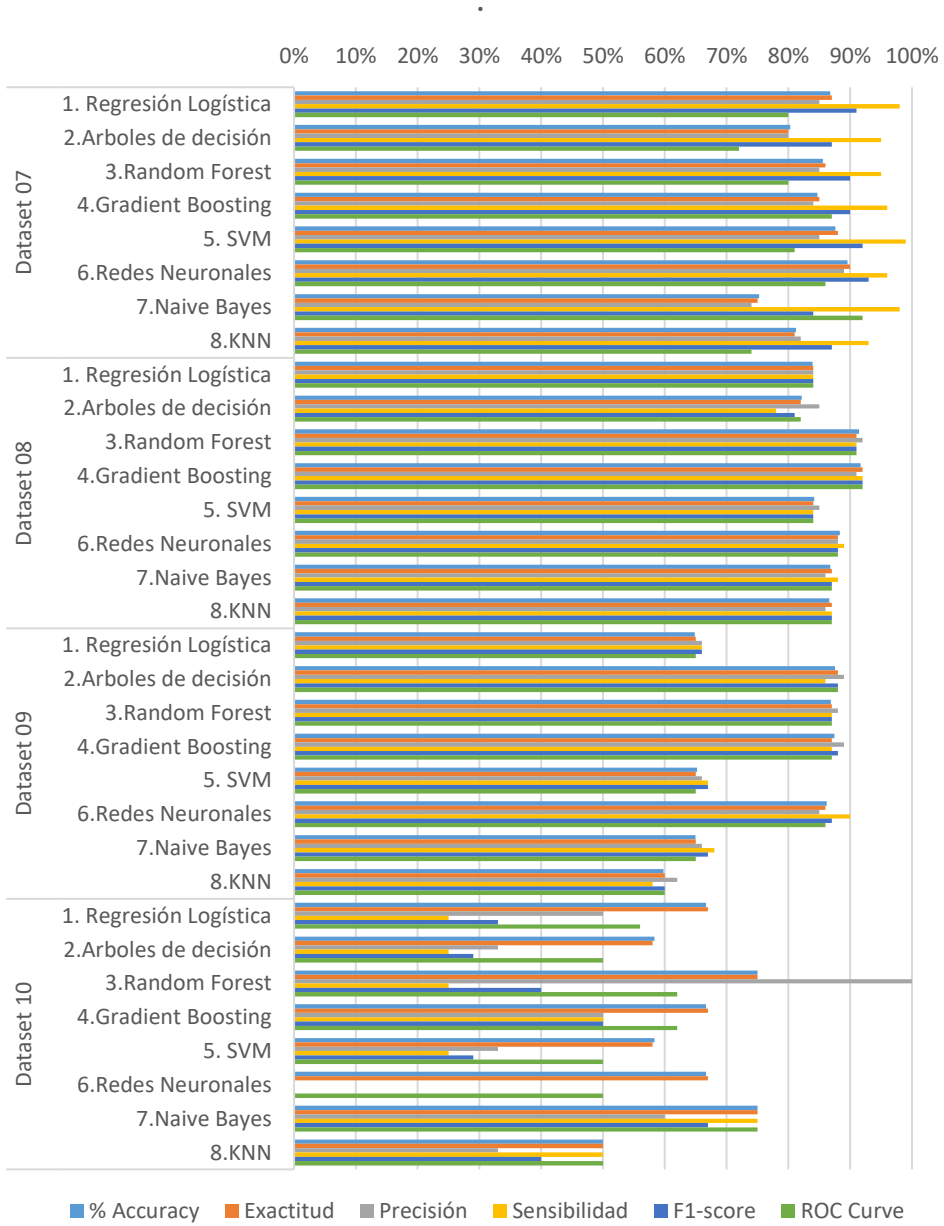


Figura 8: Normalización de Mín-Máx [0,1] & Aplicación de Algoritmo de ML - Parte 2

Normalización de Mín-Máx [-1,1] & Aplicación de Algoritmos de ML

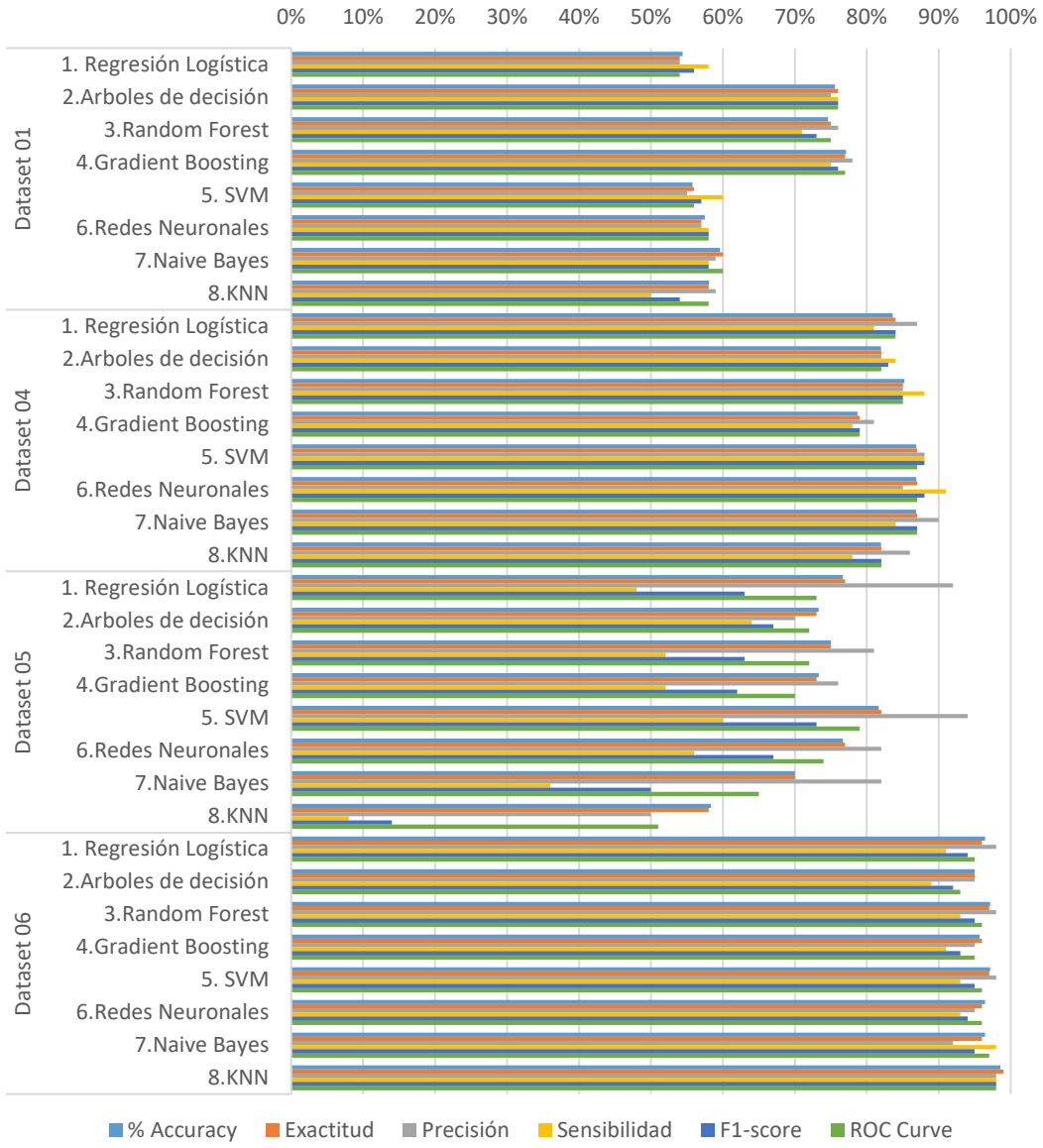


Figura 11: Normalización de Mín-Máx [-1,1] & Aplicación de Algoritmo de ML - Parte 1

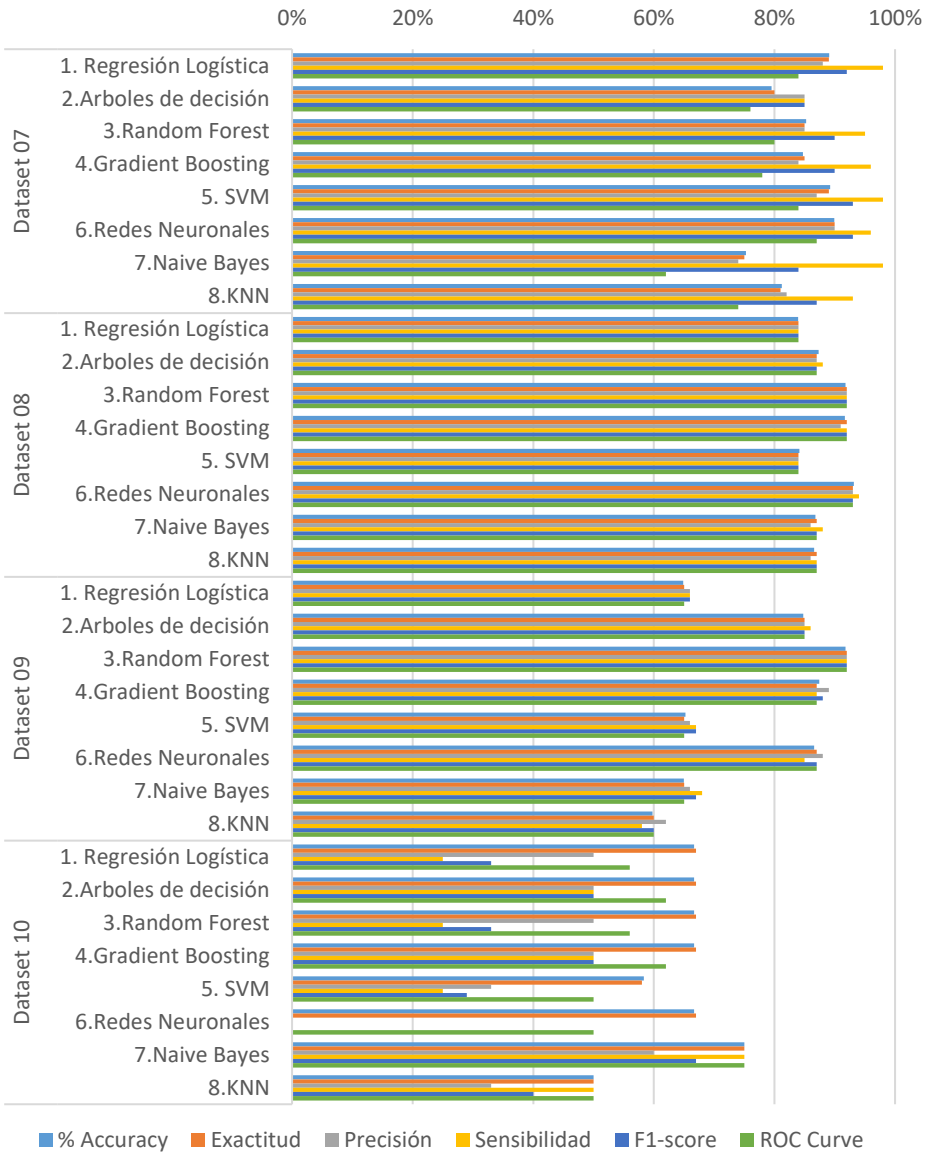


Figura 10: Normalización de Mín-Máx [-1,1] & Aplicación de Algoritmo de ML - Parte 2

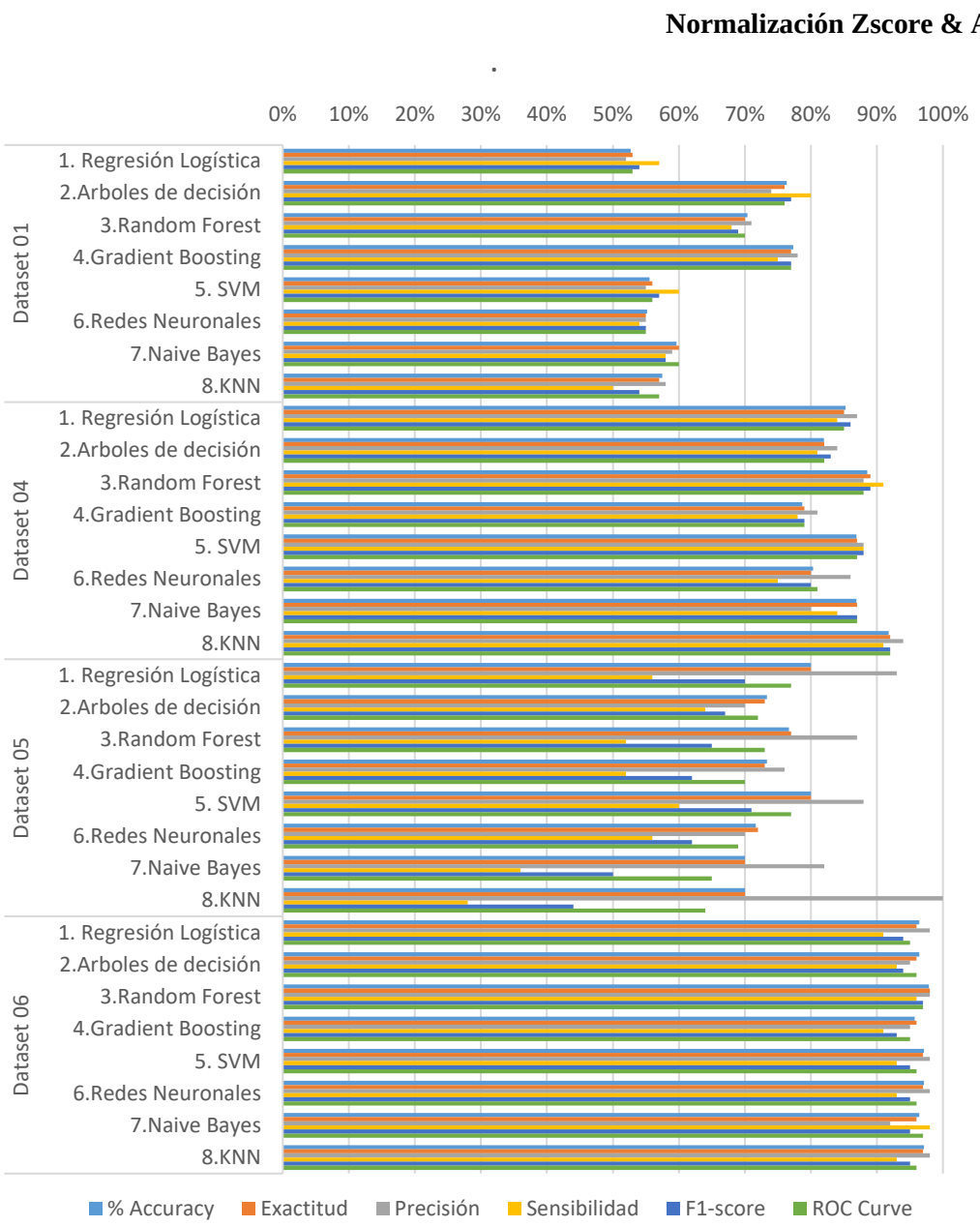


Figura 13: Normalización ZScore & Aplicación de Algoritmo de ML- Parte 1

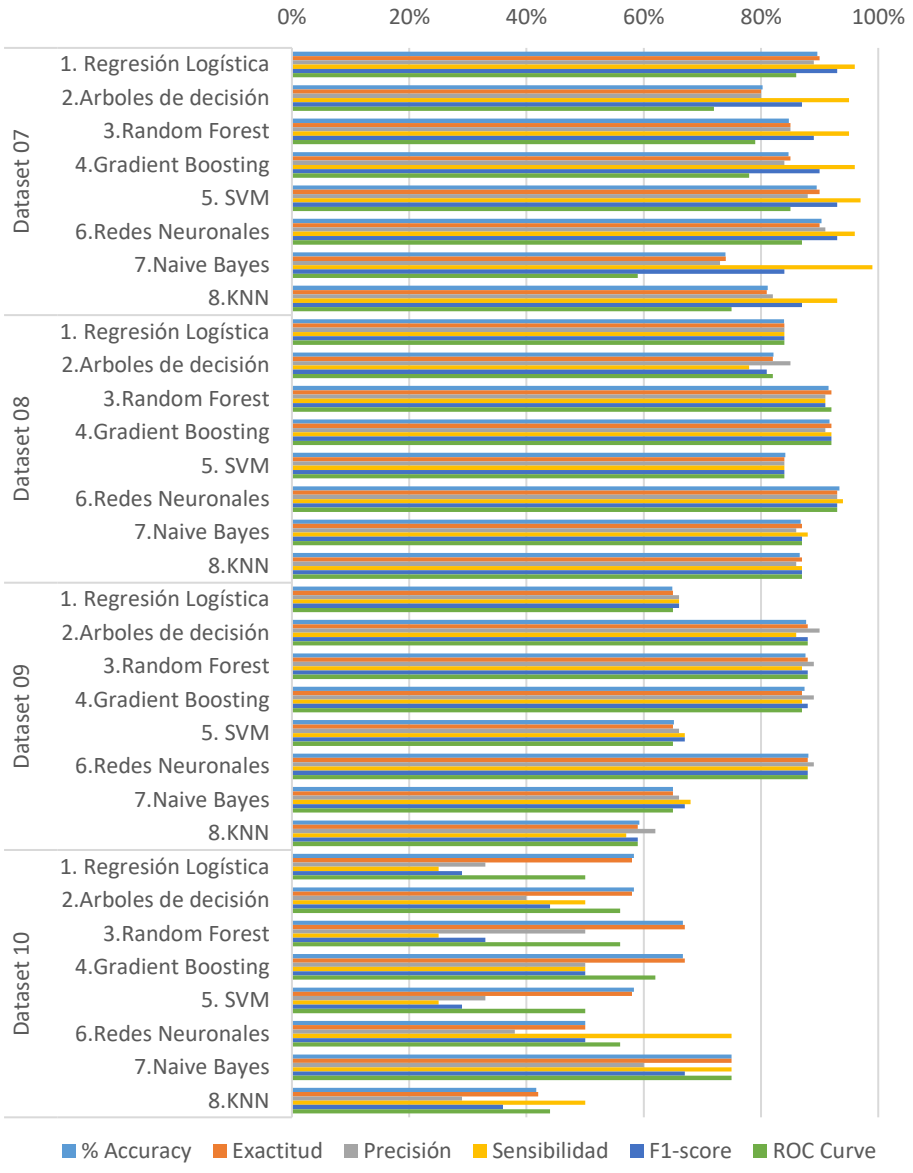


Figura 12: Normalización ZScore & Aplicación de Algoritmo de ML - Parte 2

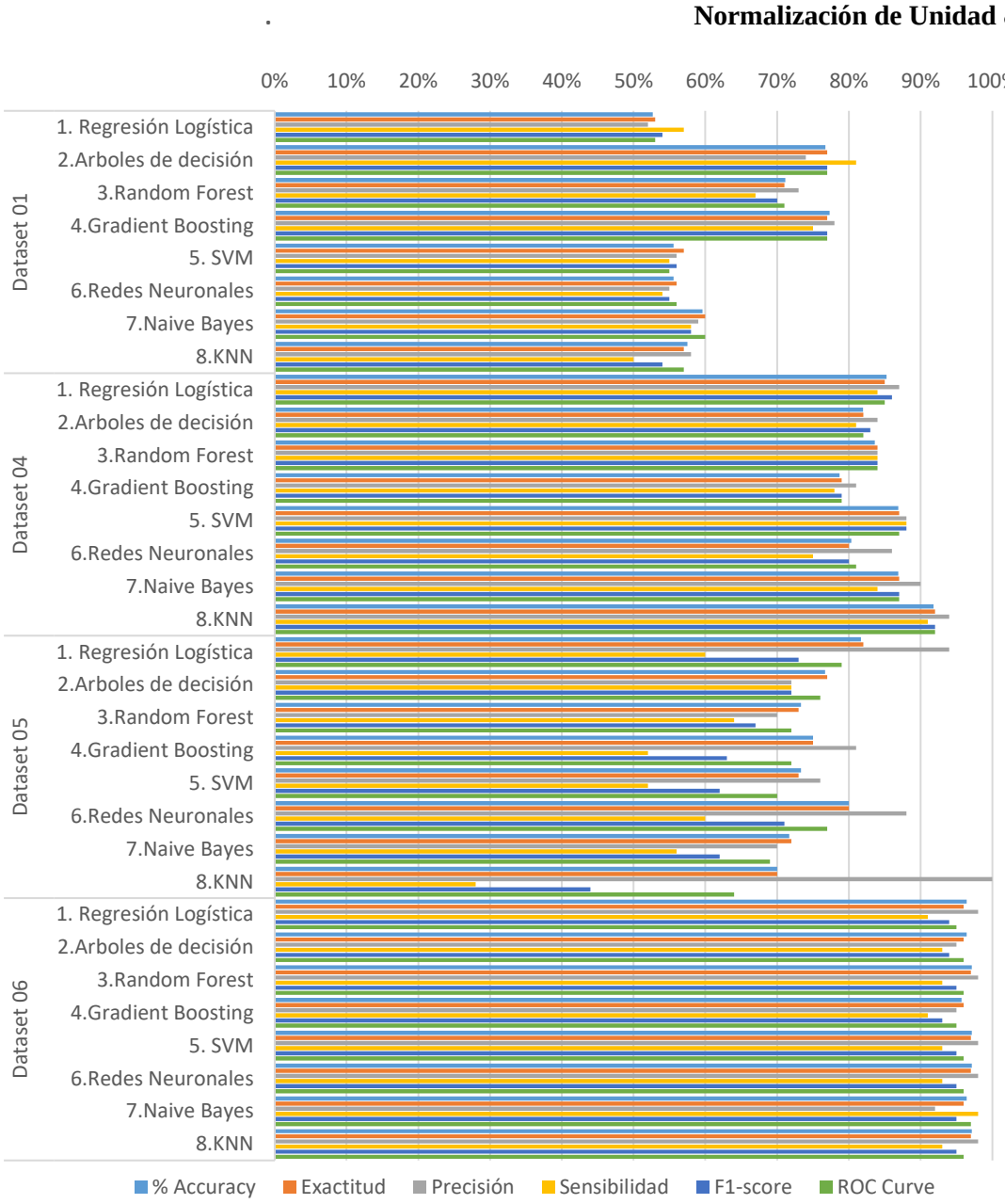


Figura 15: Normalización de Unidad & Aplicación Algoritmos de ML - Parte 1

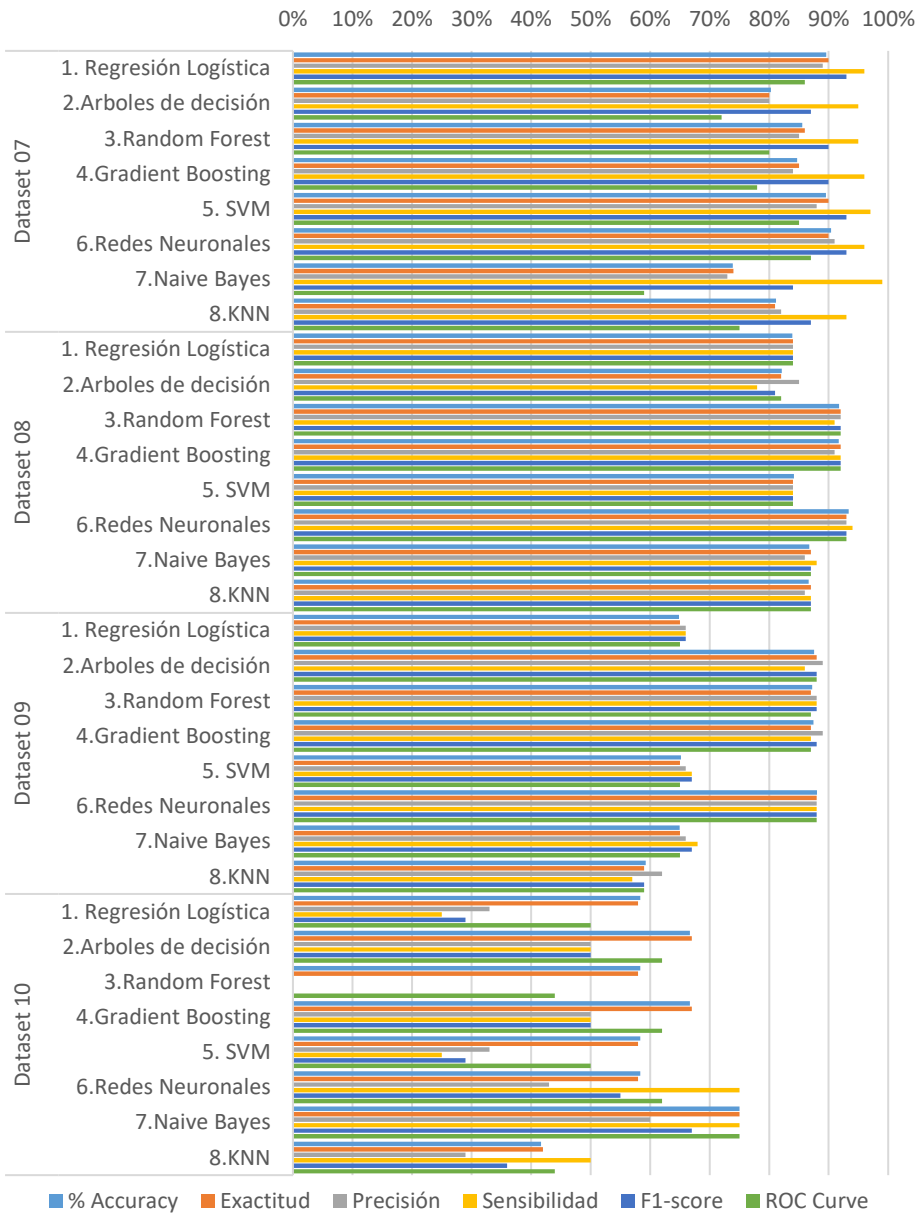


Figura 14: Normalización de Unidad & Aplicación Algoritmos de ML- Parte 2

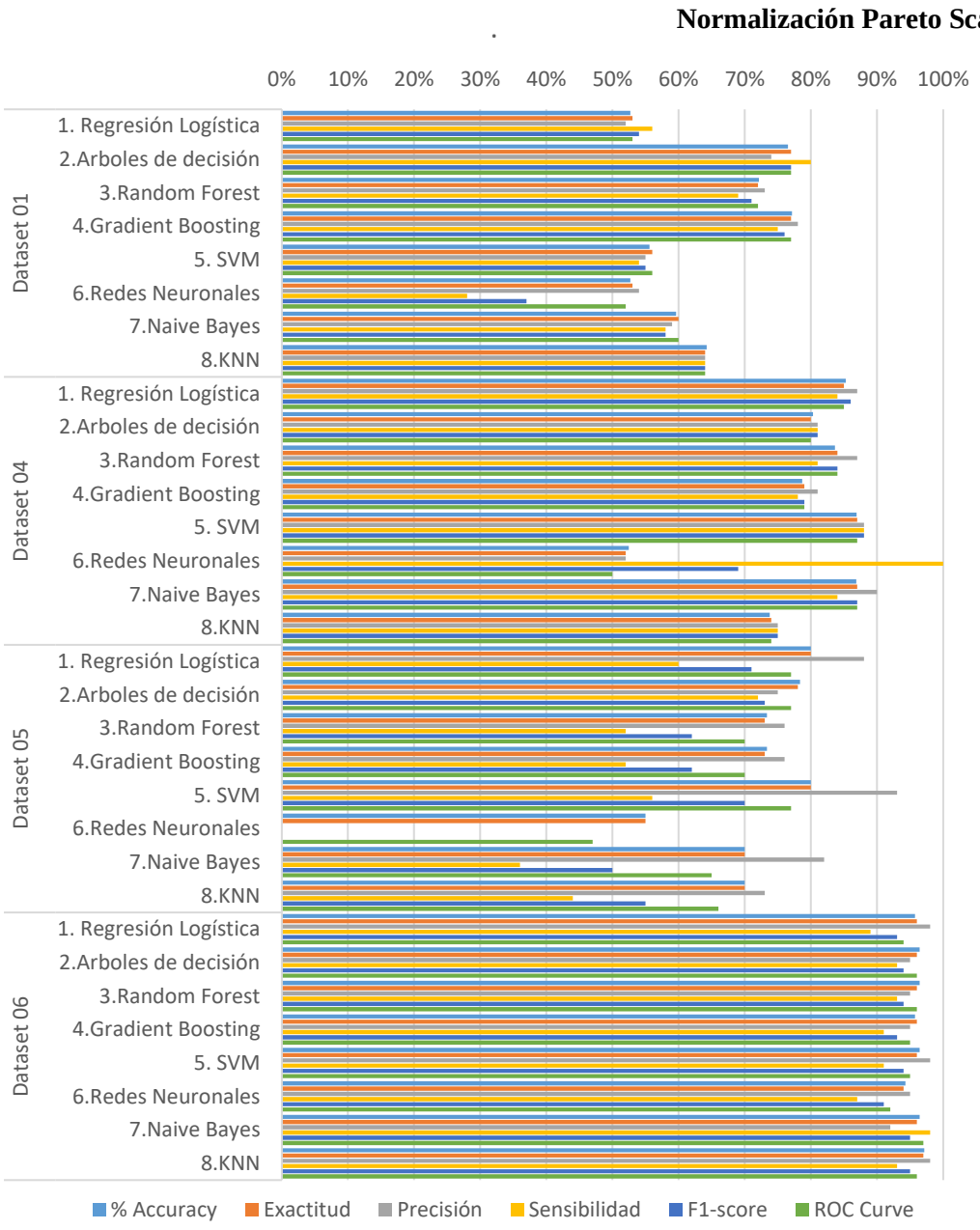


Figura 17: Normalización PScaling & Aplicación de Algoritmo de ML - Parte 1

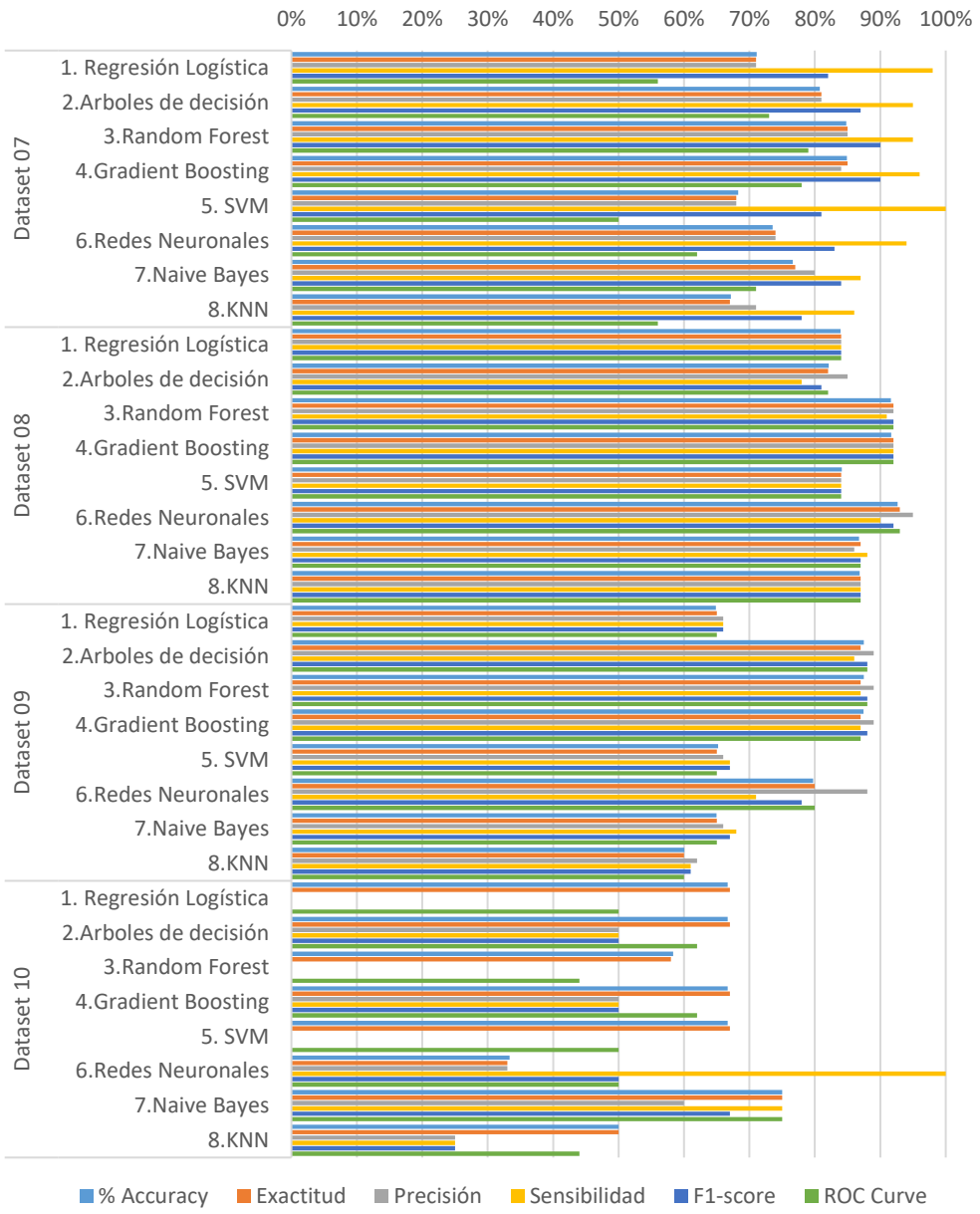


Figura 16: Normalización PScaling & Aplicación de Algoritmo de ML - Parte 2

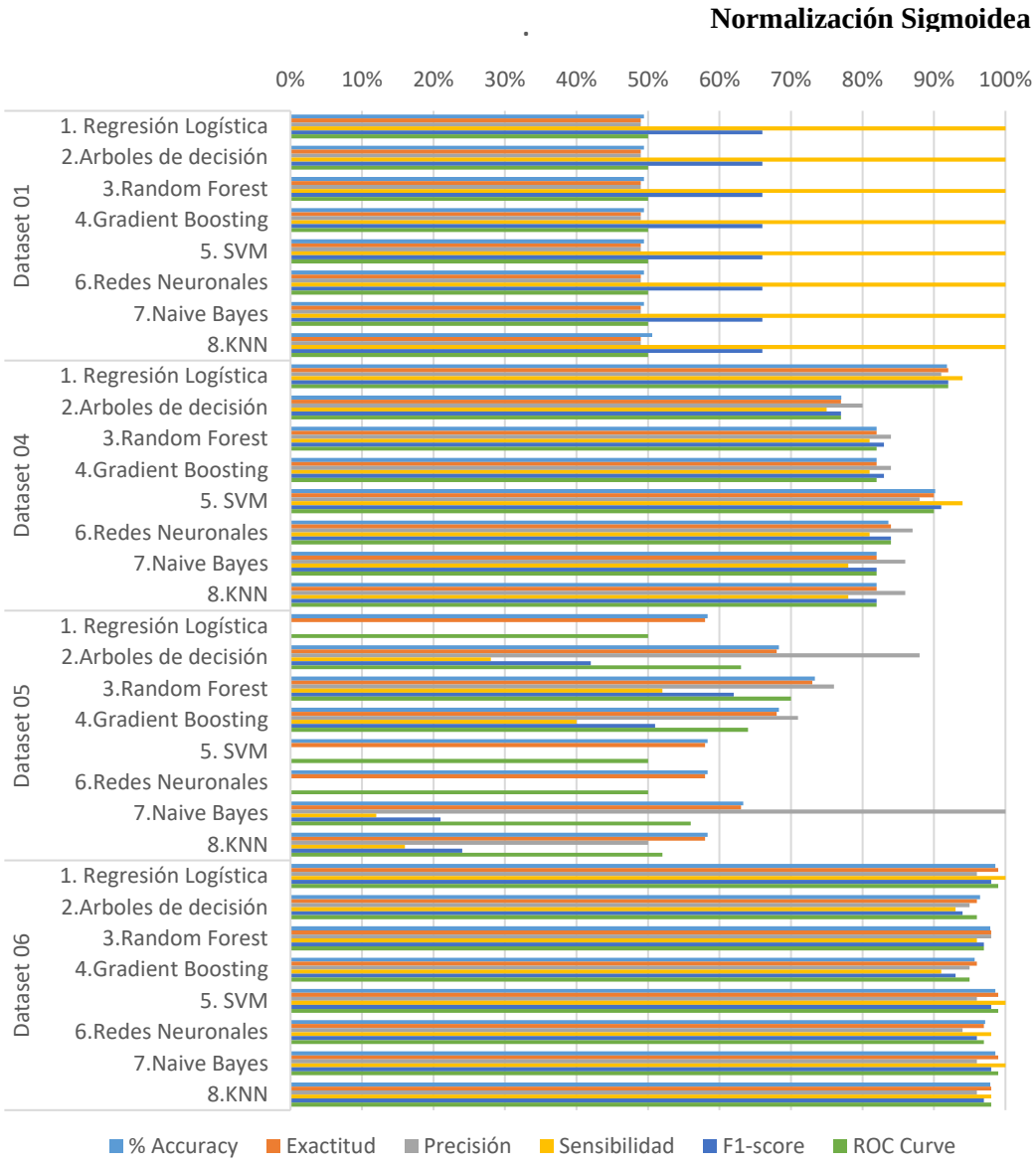


Figura 18: Normalización Sigmoidea & Aplicación de Algoritmo de ML - Parte 1

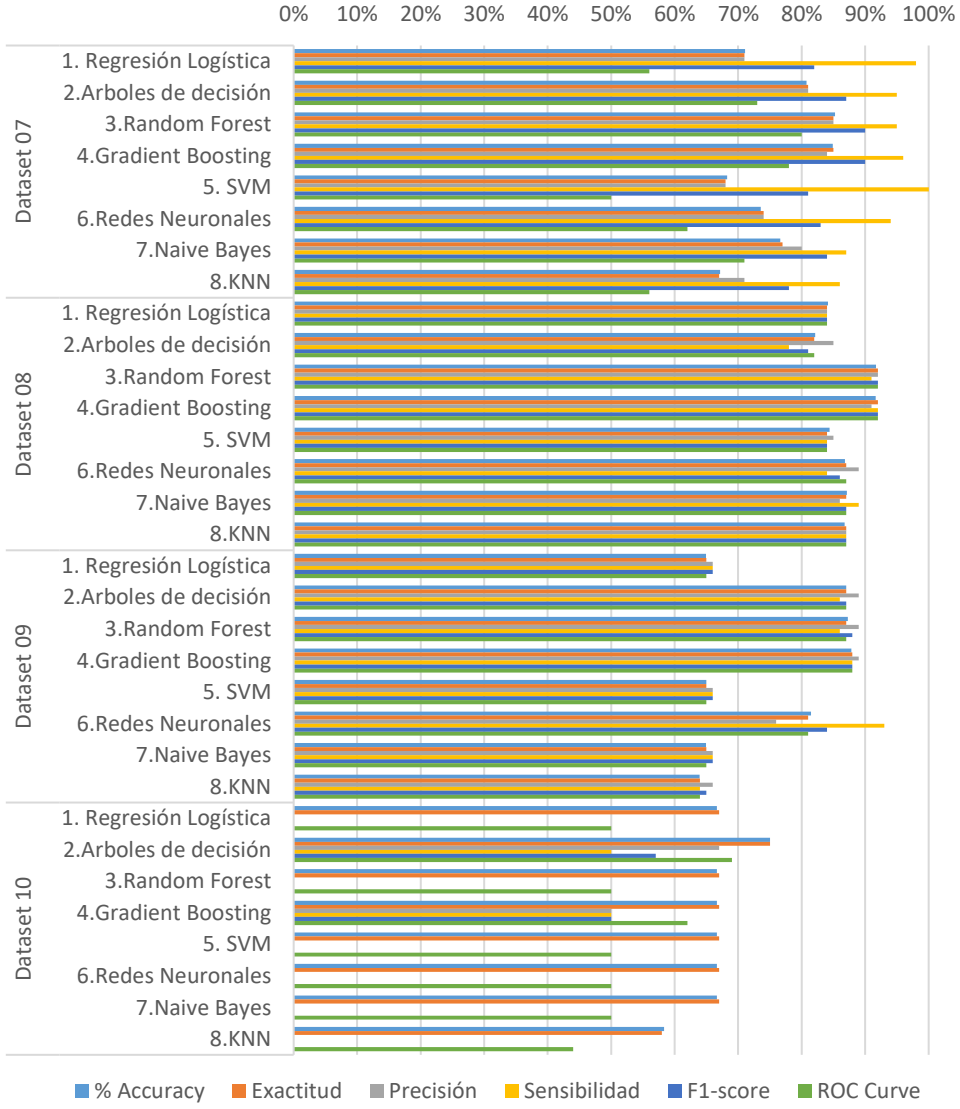


Figura 19: Normalización Sigmoidea & Aplicación de Algoritmo de ML - Parte 2

De igual manera, para ambos conjuntos de datos de regresión, los resultados se clasificaron en tres categorías. En la primera categoría, se consideraron modelos aceptables aquellos con métricas (MSE, MAE, RMSE) con valores por debajo de <= 5 y R^2 >= 90. La segunda categoría incluye modelos con métricas por debajo de <= 15 y R^2 >= 30, los cuales arrojaron resultados mixtos. Por último, la tercera categoría abarca modelos con métricas <= 20 y R^2 >= 20, indicando un rendimiento bajo. A continuación, se presentan las gráficas correspondientes a cada una de estas categorías.

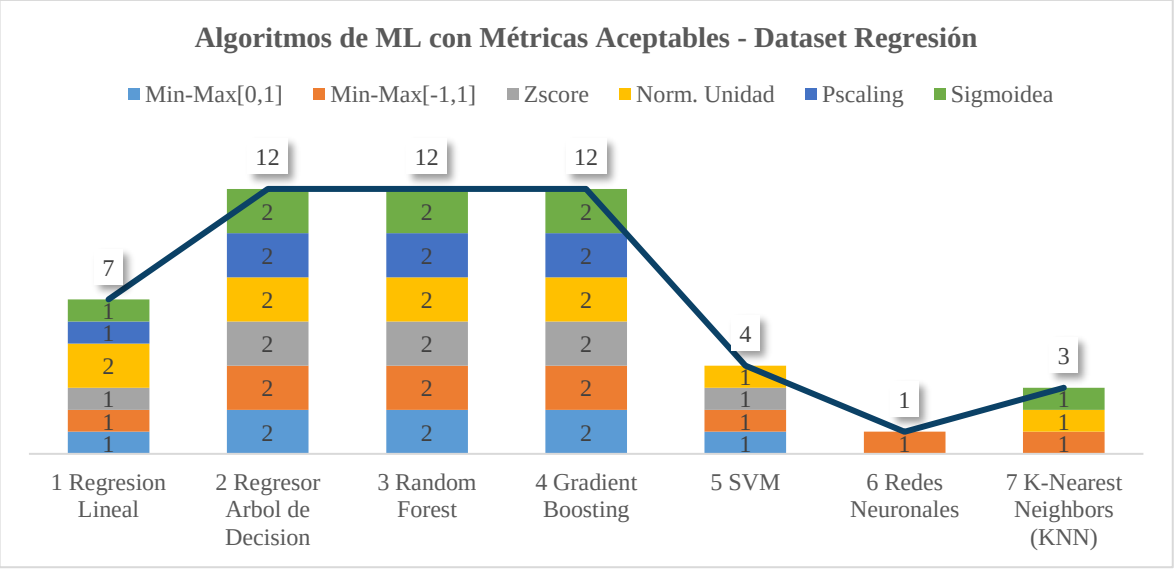


Figura 20: Algoritmos de ML con métricas aceptables - Dataset Regresión.

De acuerdo a la gráfica (Figura 21) se observa a los algoritmos que lograr métricas aceptables son: Regresor Árbol de Decisión, Random Forest y Gradient Booting.

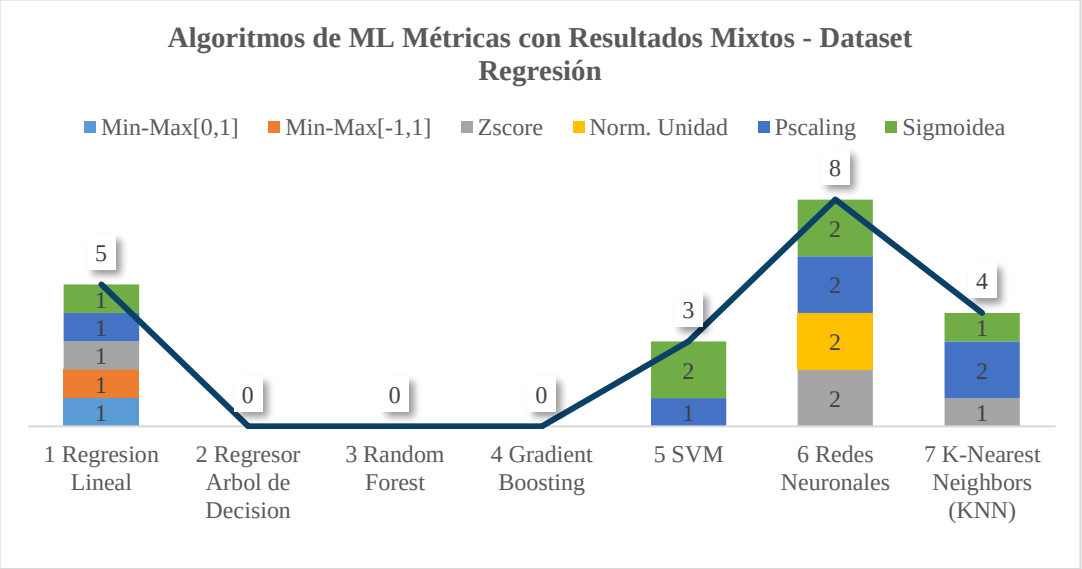


Figura 21: Algoritmos de ML con métricas Mixtos - Dataset Regresión.

En la figura 22 se muestra a los algoritmos que tuvieron resultados mixtos son: Regresión Lineal, Redes Neuronales y *k*NN.

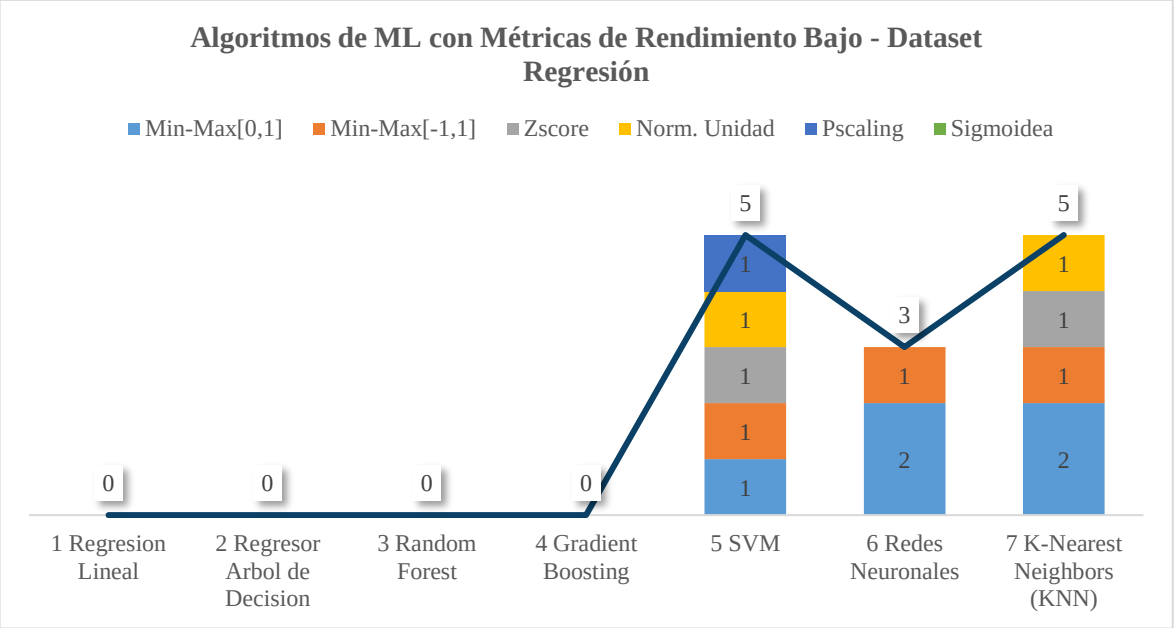


Figura 22:Algoritmo de ML con métricas Bajos - Dataset Regresión.

Las métricas de los algoritmos SVM, Redes Neuronales y *k*NN no lograron superar el umbral definido, por lo tanto, son los algoritmos no tuvieron buenos resultados.

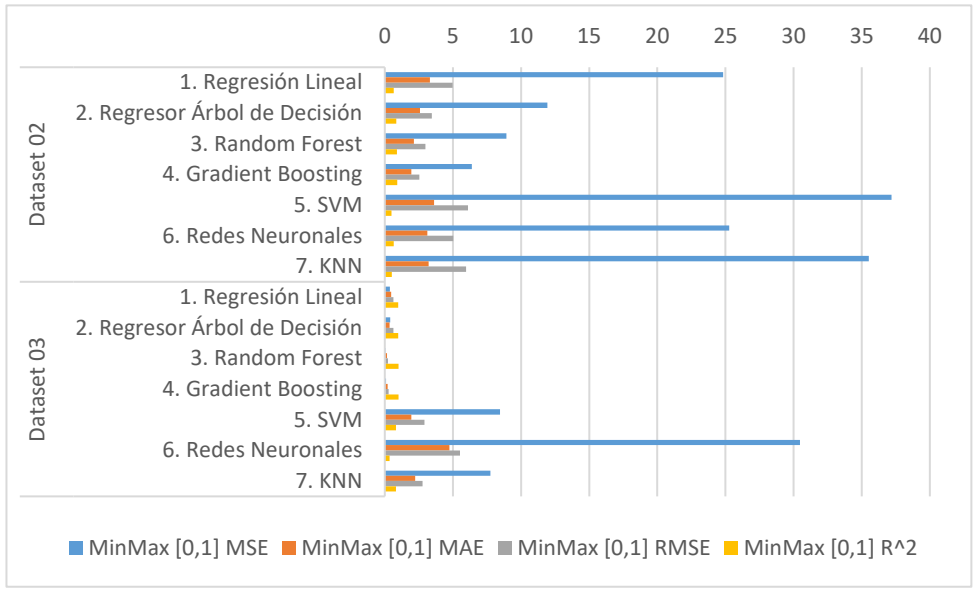


Figura 26: Normalización MinMax [0,1] & Aplicación de Algoritmo de ML–Dataset Regresión

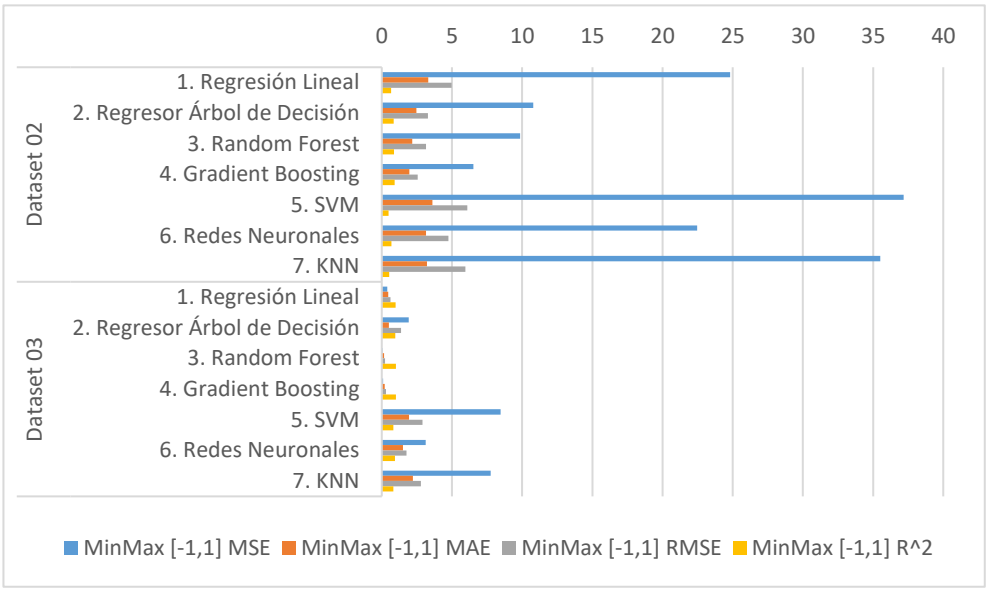


Figura 24: Normalización MinMax [-1,1] & Aplicación de Algoritmo de ML–Dataset Regresión

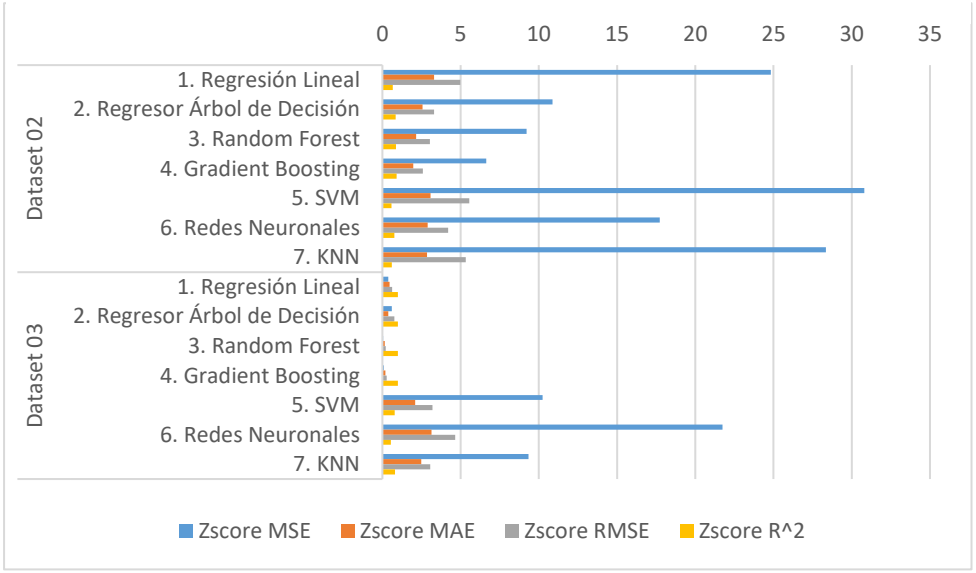


Figura 23: Normalización Zscore & Aplicación de Algoritmo de ML – Dataset Regresión

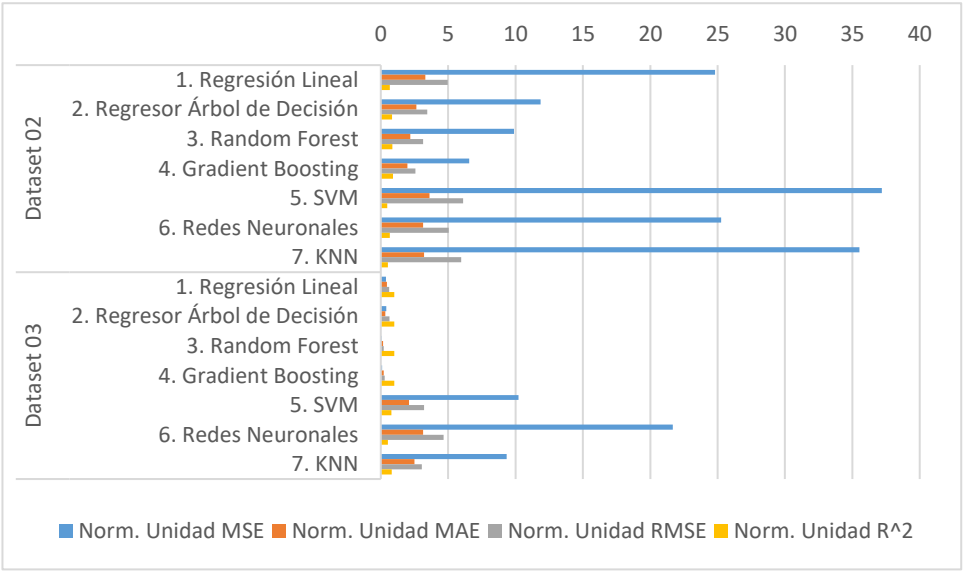


Figura 25: Normalización Unidad & Aplicación de Algoritmo de ML – Dataset Regresión

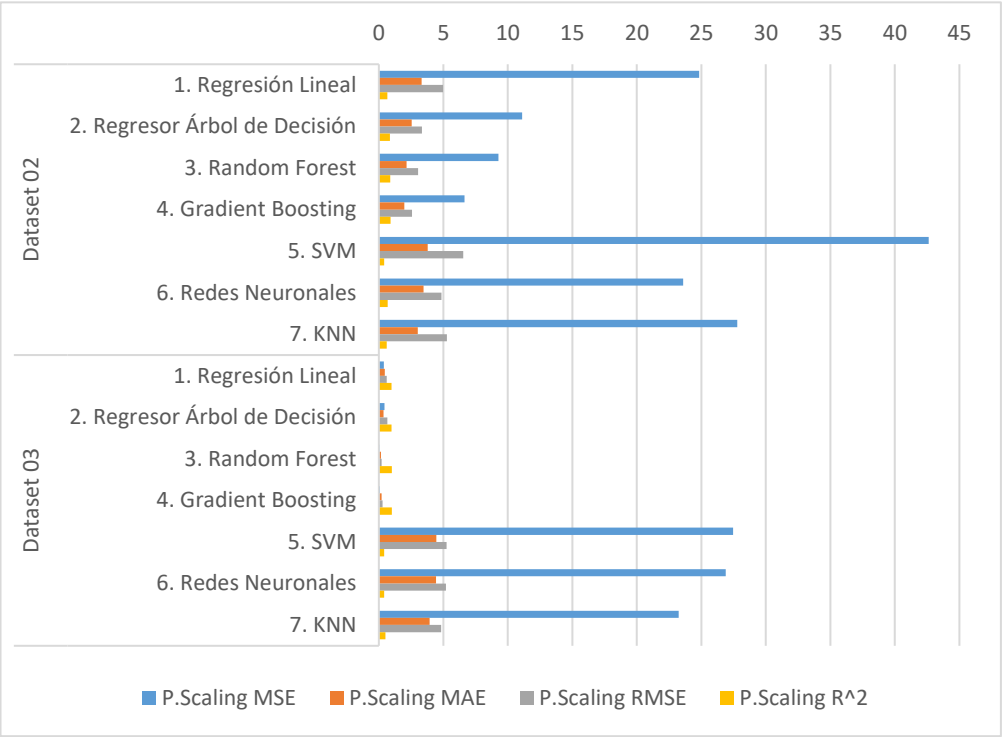


Figura 28: Normalización PScaling & Aplicación de Algoritmo de ML – Dataset Regresión

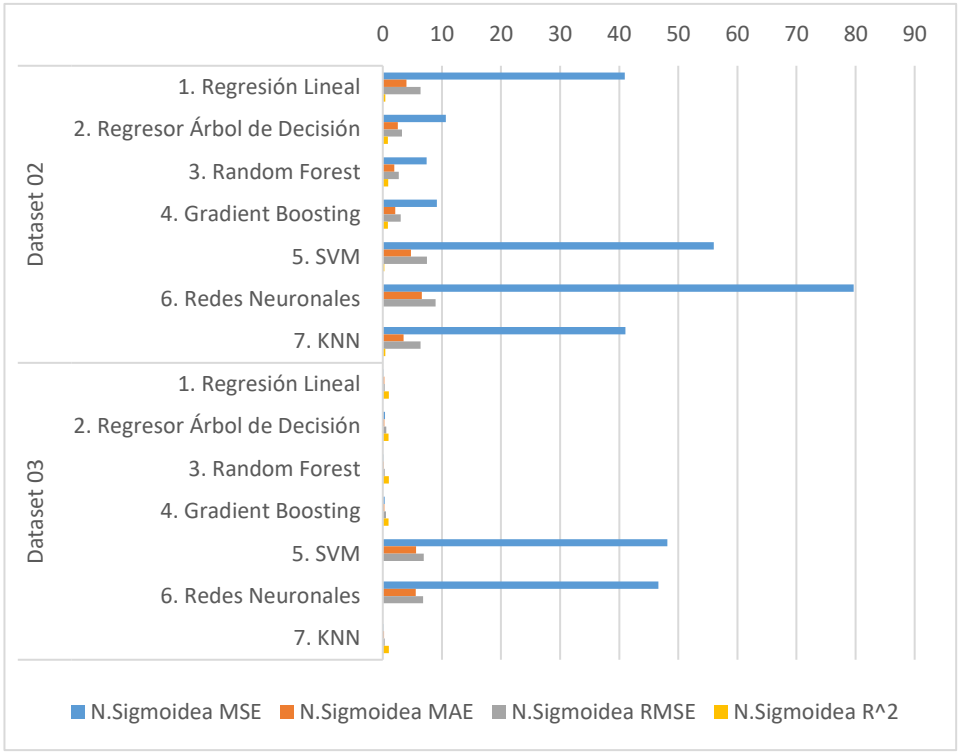


Figura 27: Normalización Sigmoidea & Aplicación de Algoritmo de ML – Dataset Regresión

6. CONCLUSIONES

En conclusión, el presente estudio comparativo sobre el impacto de la normalización de datos en la precisión de los algoritmos de Machine Learning ha proporcionado información valiosa sobre el comportamiento de diversos algoritmos frente a diferentes métodos de normalización. Los resultados obtenidos permiten extraer conclusiones relevantes para la selección y optimización de los algoritmos en el ámbito de la clasificación y regresión de datos.

Según el estudio realizado, se evaluó el rendimiento de ocho algoritmos de aprendizaje automático: Regresión Logística, Árboles de decisión, Random Forest, Gradient Boosting, SVM, Redes Neuronales, Naïve Bayes y *k*NN, en ocho conjuntos de datos de clasificación. Nuestros resultados muestran que Random Forest es el algoritmo que con mayor frecuencia al superar el umbral del 80% en las métricas de rendimiento evaluadas (% Accuracy, Exactitud, Precisión, Sensibilidad, F1-score y Curva ROC). En particular, Random Forest superó este umbral en 25 ocasiones, seguido de SVM y Redes Neuronales con 22 ocasiones cada uno. La Regresión Logística, Naïve Bayes y Árboles de Decisión también alcanzaron el umbral del 80% en 21, 18 y 18 ocasiones, respectivamente. En contraste, *k*NN solo superó este umbral en 14 ocasiones, siendo el algoritmo con el rendimiento más bajo entre los 7 algoritmos restantes.

En el grupo de algoritmos que obtuvieron métricas de rendimiento entre el 60% y el 79%, se observan algunas tendencias notables. En particular, Árboles de Decisión es el algoritmo que más frecuentemente superó el umbral del 60%, con 18 ocasiones. En segundo lugar, Gradient Boosting obtuvo 15 ocasiones por encima del umbral. Luego, un trío de algoritmos, Random Forest, Naïve Bayes y *k*NN, también superó el umbral en 11 ocasiones cada uno. Los algoritmos de Regresión Logística, SVM y Redes Neuronales alcanzaron el umbral en 9, 8 y 7 ocasiones, respectivamente. Es importante destacar que estos últimos tres algoritmos presentan una variabilidad moderada en su frecuencia de cumplimiento.

En el grupo de algoritmos que obtuvieron métricas de rendimiento por debajo del 59%, *k*NN es el algoritmo que más frecuentemente mostró un rendimiento por debajo del umbral, con 19 ocasiones. Le sigue Redes Neuronales con 17 ocasiones, seguido de un grupo de cuatro algoritmos, Regresión Logística, SVM, Naïve Bayes y Gradient Boosting, con 16 ocasiones cada uno. Random Forest y Árboles de Decisión mostraron un rendimiento por debajo del umbral en 10 y 9 ocasiones, respectivamente.

De igual manera, al realizar un análisis de los resultados en función de las técnicas de normalización, se llega a la conclusión de que la estrategia de normalización que se incorporó de manera más fluida a los algoritmos y logró superar sus métricas en un porcentaje superior al 80%, es la técnica de Mín-Máx [0,1] (en 30 ocasiones), seguida por Max [-1,1] (en 28 ocasiones) y Normalización de Unidad (en 29 ocasiones). Por otro lado, se observa que las técnicas de normalización que generaron resultados variados en las métricas de evaluación de sus algoritmos fueron Zscore (en 44 ocasiones), Pscalin (en 21 ocasiones), es decir, sus resultados de métricas están por debajo del 79%, y finalmente el algoritmo Sigmoidea (en 71 ocasiones), de las cuales 32 se ubicaron por debajo del umbral del 60%.

Además, en los dos conjuntos de datos de regresión, los algoritmos que tuvieron un mejor comportamiento frente a las 5 técnicas de normalización (Mín-Máx, Z-Score, Normalización de Unidad, Pareto Scaling y Sigmoidea) fueron Regresión Lineal, Regresor Árbol de Decisión, Random Forest y Gradient Boosting, con mejores resultados en sus métricas MSE, MAE, RMSE y R^2, mientras que los algoritmos SVM, Redes Neuronales, Regresión Lineal y *k*NN tuvieron resultados mixtos.

En resumen, los resultados de este estudio resaltan la importancia de la normalización de datos en los algoritmos de Machine Learning y su impacto en la precisión de la clasificación y regresión. La elección adecuada del método de normalización puede contribuir significativamente a mejorar el rendimiento y la precisión de los algoritmos. Estos hallazgos proporcionan un marco de referencia valioso para futuras investigaciones y aplicaciones prácticas, facilitando la selección de técnicas de normalización adecuadas en función de las características y requisitos específicos de los conjuntos de datos.

7. REFERENCIAS

[1] Andreas Müller and Sarah Guido, Book: *Introduction to Machine Learning with Python, a Guide for Data Scientists*, Oreilly. 2017.

[2] Kevin P. Murphy, Book: *Machine Learning a Probabilistic Perspective*, The MIT Press. London, England: Massachusetts Instituteof Technology, 2012.

[3] Rudolph Russell, Book: *Machine Learning Guia Paso a Paso para Implementar Algoritmos de Machine Learning con Python*. 2018.

[4] Dipanjan Sarkar, Raghav Bali, and Tushar Sharma, Book: *Practical Machine Learning with Python A Problem-Solver's Guide to Building Real-World Intelligent Systems*. New York: Springer Science + Business Media, 2018. doi: 10.1007/978-1-4842-3207-1.

[5] Sarat Nayak C, Misra Bijan B, and Behera S, Scientific Magazine: "Impact of Data Normalization on Stock Index Forecasting," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 6, pp. 257–269, 2014

[6] Robert A van den Berg, Huub CJ Hoefsloot, Johan A Westerhuis, Age K Smilde, and Mariët J van der Werf, Research Article: "Centering, Scaling, and Transformations: Improving the Biological Information Content of Metabolomics Data," *BMC Genomics*, vol. 7, Jun. 2006, doi: 10.1186/1471-2164-7-142.

[7] Sruthi Namburi, Master's Thesis: "Logistic Regression with Conjugate Gradient Descent for Document Classification," *Jawaharlal Nehru Technology University (JNTU)*, India, 2013.

[8] Ahmed Abdelkarim Eldud Omer, Master's Thesis: "Prediction of Protein Secondary Structure using Binary Classification Trees, Naive Bayes Classifiers and the Logistic Regression Classifier," *Rhodes University*, 2015.

[9] Jehad Ali, Rehanullah Khan, Nasir Ahmad, and Imran Maqsood, Scientific Magazine: "Random Forests and Decision Trees," *IJCSI International Journal of Computer Science Issues*, Vol. 9, Issue 5, No 3, September 2012.

[10] Robin Genuer, Jean-Michel Poggi, Christine Tuleau-Malot, and Nathalie Villa-Vialaneix, Research Article: "Random Forests for Big Data," Mar. 2015.

[11] Stefanos Fafalios, Pavlos Charonyktakis, and Loannis Tsamardinos, Research Article: "Gradient Boosting Trees," Apr. 2020

[12] Llew Mason, Peter Bartlett, Jonathan Baxter, and Marcus Frean, Research Article: "Boosting Algorithms as Gradient Descent".

[13] D. Zhang and A. Wulamu, Scientific Magazine: "Improvement of Support Vector Machine Algorithm in Big Data Background" *Mathematical Problems in Engineering, Hindawi, School of Computer and Communication Engineering, University of Science and Technology Beijing*, Beijing, China, Vol 2021, pp 1024-123X, Juny 2017, doi: https://doi.org/10.1155/2021/5594899.

[14] Mariette Awad and Rahul Khanna, Scientific Magazine: "Support Vector Machines for Classification," *Efficient Learning Machines*, pp. 39–66, 2015, doi: 10.1007/978-1-4302-5990-9_3.

[15] Fernando Bueno, Bachelor's Tesis: "Redes Neuronales: Entrenamiento y Comportamiento," *Universidad Complutense de Madrid, Madrid*, 2019.

[16] Geoffrey I. Webb, Scientific Magazine: "Naïve Bayes*" Encyclopedia of Machine Learning*, Springer US, Boston, MA 2010, pp. 713-714, SN 978-0-387-30164-8, doi: https://doi.org/10.1007/978-0-387-30164-8_576.

[17] Angel Urbano Romeu, Bachelor's Thesis: “Emotion Recognition Based on the Speech, Using a Naive Bayes Classifier,” *Escola Tècnica Superior d’Enginyeria de Telecomunicació de Barcelona*, 2016.

[18] I. Wickramasinghe and H. Kalutarage. Scientific Magazine: “Naive Bayes: Applications, Variations and Vulnerabilities: A Review of Literature with Code Snippets for Implementation” Springer Link, Soft Comput, Vol 25, pp. 2277–2293, September 2021, doi: https://doi.org/10.1007/s00500-020-05297-6

[19] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, Scientific Magazine: “Efficient kNN classification with different numbers of nearest neighbors,” *IEEE Trans Neural Netw Learn Syst*, vol. 29, no. 5, pp. 1774–1785, May 2018, doi: 10.1109/TNNLS.2017.2673241.

[20] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, Scientific Magazine: “KNN model-based approach in classification,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2888, pp. 986–996, 2003, doi: 10.1007/978-3-540-39964-3_62.

[21] Daniel Victor Jaramillo Muñoz, Master's Thesis: “Evaluacion de Modelos de Machine Learning para la Prediccion de Crımenes en la Ciudad de Medellın,” *Universidad Nacional de Colombia, Medellin*, 2021.

[22] Peter Flach, Scientific Magazine: "Performance Evaluation in Machine Learning: The Good, The Bad, The Ugly and The Way Forward" *Intelligent Systems Laboratory, University of Bristol, UK*, vol. 33, N°.01, pp 9808-9814, July. 2017.

[23] Anita Desiani, Adinda Ayu Lestari, M Al-Ariq, Ali Amran, and Yuli Andriani, Scientific Magazine: “Comparison of Support Vector Machine and K-Nearest Neighbors in Breast Cancer Classification,” *Pattimura International Journal of Mathematics (PIJMath)*, vol. 1, no. 1, pp. 33–42, May 2022, doi: 10.30598/pijmathvol1iss1pp33-42.

[24] I. Izonin, R. Tkachenko, N. Shakhovska, B. Ilchyshyn, and K. K. Singh, Scientific Magazine: “A Two-Step Data Normalization Approach for Improving Classification Accuracy in the Medical Diagnosis Domain,” *Mathematics*, vol. 10, no. 11, Jun. 2022, doi: 10.3390/math10111942.

[25] Ivan Miguel Pires, Faisal Hussain, Nuno M Garcia, Petre Lameski, and Eftim Zdravevski, Scientific Magazine: “Homogeneous Data Normalization and Deep Learning: A Case Study in Human Activity Classification,” *Future Internet*, vol. 12, no. 11, pp. 1–14, Nov. 2020, doi: 10.3390/fi12110194.

[26] Markelle Kelly, Rachel Longjohn, Kolby Nottingham, The UCI Machine Learning Repository, https://archive.ics.uci.edu.

[27] Web Page Kaggle.com, “Kaggle: Your Machine Learning and Data Science Community.” https://www.kaggle.com/.

[28] Web Page OpenML A worldwide machine learning lab, “OpenML A worldwide machine learning lab.” https://openml.org/.

[29] Isabelle Guyon, Steve R Gunn, Asa Ben-Hur, Gideon Dror, “Madelon Dataset,” *Result analysis of the NIPS 2003 feature selection challenge*, Distributed by OpenML, 2004.

[30] Onur Yildirim “Boston house price Dataset,” *Servicio del Censo de EE. UU*, Distributed by OpenML, 2022.

[31] Roger W Johnson “Body fat Percentage Estimates Dataset,” *UCI - Irvine Machine Learning Repository, TunedIT*, Distributed by UCI, 2014.

[32] Rashik Rahman “Heart Attack Analysis & Prediction Dataset” Distributed by Kaggle, Aug. 2021.

[33] Larxel “Heart Failure Prediction Dataset” Distributed by Kaggle, Aug. 2020.

[34] Dr. William H. Wolberg “Breast Cancer Wisconsin Dataset,” *UCI - Irvine Machine Learning Repository, University of Wisconsin*, Distributed by UCI, 1995.

[35] “Elevators Dataset” Distributed by OpenML, Oct. 2014.

[36] “Fried Dataset” Distributed by OpenML, Oct. 2014.

[37] “Puma32H Dataset” Distributed by OpenML, Oct. 2014.

[38] “Tumours of the central nervous system Dataset” Distributed by OpenML, Nov. 2014.