

Configuración de hiperparámetros mediante algoritmos de optimización: Aplicación en la predicción de enfermedades cardiovasculares

Eduardo Sánchez-Jiménez, Yasmín Hernández,
Javier Ortiz-Hernández, Alicia Martínez-Rebollar,
Hugo Estrada-Esquivel

Centro Nacional de Investigación y Desarrollo Tecnológico,
Morelos,
México

{m22ce005, yasmin.hp, javier.oh, alicia.mr,
hugo.ee}@cenidet.tecnm.mx

Resumen. El desarrollo de modelos predictivos involucra la selección de un algoritmo de aprendizaje automático, los cuales tienen un conjunto de hiperparámetros que controlan su entrenamiento y desempeño. Los valores adecuados para los hiperparámetros dependen de la naturaleza de los datos, por lo que es necesario definirlos para cada modelo de aprendizaje que se construya. Se plantean las técnicas de optimización como mecanismos de configuración de hiperparámetros para buscar una estructura eficiente para el entrenamiento de los algoritmos de aprendizaje. Este artículo presenta la aplicación de las técnicas *Grid Search*, *Random Search* y *Bayesian Optimization* para configurar los hiperparámetros de los algoritmos *Random Forest*, *Support Vector Machine* y *XGBoost*. Se busca construir modelos de predicción de enfermedades cardiovasculares, por lo que se utilizó el conjunto de datos *Cleveland* de la Universidad de California en Irvine. Se identificó que el modelo *Random Forest* logró el mejor rendimiento al obtener un 90% en la métrica *Precision*, utilizando el optimizador bayesiano.

Palabras clave: Ajuste de hiperparámetros, enfermedades cardiovasculares, técnicas de optimización.

Hyperparameter Configuration by Optimization Algorithms: Application in the Prediction of Cardiovascular Disease

Abstract. The development of predictive models involves the selection of a machine learning algorithm, which has a set of hyperparameters that controls its training and performance. The appropriate values for the hyperparameters depend on the nature of the data, so it is necessary to define them for each learning model to be built. Optimization techniques are proposed as a hyperparameter configuration mechanism to find an efficient structure for the training of learning algorithms. This paper presents the application of Grid Search, Random Search and Bayesian Optimization techniques to configure the hyperparameters of

Random Forest, Support Vector Machine and XGBoost algorithms. The aim is to build efficient prediction models for cardiovascular disease, so we used the Cleveland dataset from the University of California, Irvine. It was identified that the *Random Forest* model achieved the best performance by obtaining 90% in the *Precision* metric, using the Bayesian optimizer.

Keywords: Hyperparameter tuning, cardiovascular disease, optimization techniques.

1. Introducción

La predicción de enfermedades cardiovasculares (EC) es un área de interés en la medicina debido a su alta prevalencia y mortalidad en todo el mundo. Los modelos de aprendizaje automático se han implementado con éxito en la predicción de EC, ya que pueden considerar múltiples factores de riesgo y proporcionar una evaluación precisa del riesgo cardiovascular [1].

El diseño de modelos predictivos [2, 5] se considera un proceso largo y complejo que implica seleccionar el mejor algoritmo y determinar la configuración de los hiperparámetros [6]. Los hiperparámetros son importantes porque controlan el comportamiento del algoritmo en la fase de entrenamiento lo que genera un impacto significativo en el rendimiento de los modelos.

Por lo general, en la construcción de modelos predictivos se omite el ajuste de valores para los hiperparámetros, lo que conlleva al uso de los algoritmos bajo una configuración predeterminada, es decir, utilizar los valores establecidos por los desarrolladores de las bibliotecas de aprendizaje automático [1]. En algunos casos, porque no se tiene conocimiento del impacto que tienen los hiperparámetros en el desempeño de los modelos, o porque no se cuentan con mecanismos de configuración para entrenar los algoritmos.

Entre las técnicas de optimización más comunes se encuentran la *Grid Search* (GS), *Random Search* (RS) y *Bayesian Optimization* (BO). En GS se realiza la exploración sistemática y se evalúan todas las combinaciones posibles de valores de hiperparámetros [2]. Por otro lado, RS, selecciona combinaciones de hiperparámetros al azar dentro de un rango predefinido [3]. BO utiliza una función de probabilidad para encontrar la mejor combinación de hiperparámetros mediante la exploración guiada de la superficie de la función objetivo [4].

En este artículo se construyen modelos para predecir EC utilizando datos de fuentes públicas, en particular el conjunto de datos *Cleveland* del repositorio de la UCI [5]. Se busca mejorar el desempeño de los modelos predictivos a partir de la configuración de los hiperparámetros mediante un enfoque de optimización.

Sin embargo, primero es importante identificar la variedad de hiperparámetros con los que cuentan los algoritmos de aprendizaje automático. Posteriormente, entender los fundamentos de la optimización en el aprendizaje automático para tener conocimiento del comportamiento general de las técnicas de optimización y poder plantear un espacio de búsqueda de acuerdo con los algoritmos que se desean modelar.

El resto del artículo se organiza de la siguiente manera: en la sección 2 se describen los fundamentos de la optimización y los elementos que conforman dicho proceso, se presentan los hiperparámetros de los algoritmos *Random Forest* (RF) [6], *Support*

Vector Machine (SVM) [7] y Extreme Gradient Boosting (XGBoost) [8] y algunas investigaciones relacionadas.

En la sección 3 se muestran los resultados del análisis exploratorio y preprocesamiento del conjunto de datos *Cleveland*.

En la sección 4 se presentan los resultados de los experimentos de optimización y se brinda un análisis de los resultados para identificar los modelos que presentan mayor porcentaje de precisión en la predicción de personas con EC. Finalmente, la sección 5 se exponen las conclusiones y se propone algunas guías para el trabajo futuro.

2. Antecedentes y trabajo relacionado

2.1. Optimización matemática

Un problema de optimización consiste en la búsqueda de valores para determinadas variables, comúnmente llamadas variables de decisión, de forma que, cumpliendo un conjunto de restricciones proporcionan el mejor valor posible para una función objetivo, que utiliza para medir el rendimiento del sistema que se estudia. Un problema de optimización con restricciones puede expresarse como [9]:

$$\min/\max_x f(x), \quad (1)$$

con respecto a:

$$g_i(x) \leq 0, i = 1, 2, \dots, m, \quad (2)$$

$$h_i(x) = 0, j = 1, 2, \dots, p. \quad x \in X, \quad (3)$$

donde $g_i(x) \leq 0$ es la función de restricción de desigualdad; $h_i(x) = 0$ es la función de restricción de igualdad y X es el dominio de la variable de decisión x . El objetivo de las restricciones es limitar los posibles valores de la solución óptima a ciertas áreas del espacio de búsqueda, denominadas región factible. Así, la región factible R_f de x puede ser representada como:

$$R_f = \{x \in X \mid g_i(x) \leq 0, h_i(x) = 0\}. \quad (4)$$

2.2. Algoritmos de aprendizaje automático supervisado

El objetivo del aprendizaje supervisado es obtener una función óptima en el modelo predictivo f^* para minimizar la función objetivo $\mathcal{L}(f(x), y)$ que calcula el error entre las etiquetas de las predicciones estimadas $f(x)$ y las etiquetas verdaderas y . El modelo predictivo óptimo f^* puede obtenerse mediante [10]:

$$f^* = \arg \min \frac{1}{n} \sum_{i=1}^n \mathcal{L}(f(x_i), y), \quad (5)$$

donde n es el número de datos de entrenamiento, x_i es el vector de características de la i -ésima instancia, y_i es la salida real correspondiente, y \mathcal{L} es el valor de la función objetivo de cada muestra.

A continuación, se describen los algoritmos que fueron considerados para el desarrollo de esta investigación. En [3], [11] se mencionan los hiperparámetros que tienen un mayor impacto en la fase de entrenamiento. En la Tabla 1 se describen los principales hiperparámetros, etiquetados con base en la biblioteca de aprendizaje automático *scikit-learn* [12].

2.2.1. Random forest

RF [6] es un algoritmo que ensambla una colección de árboles entrenados con un subconjunto de características y un subconjunto de datos tomados aleatoriamente. Cada árbol puede hacer un trabajo de predicción relativamente bueno, pero es probable que exista un sobreajuste en parte de los datos. Al construir muchos árboles, todos los cuales funcionan bien y se ajustan de diferentes maneras, podemos reducir la cantidad de sobreajuste promediando sus resultados [13].

2.2.2. Support vector machine

El objetivo de SVM es encontrar un hiperplano de separación entre las clases cuando estas no pueden separarse linealmente. El hiperplano de separación es un plano que divide el espacio de características en dos partes, una para cada clase. La distancia entre el hiperplano y los puntos a la frontera con cada clase (vectores de soporte) se llama margen, y el mejor hiperplano es aquel que presenta mayor margen [7].

2.2.3. Extreme gradient boosting

XGBoost [8] genera modelos secuenciales y utiliza la técnica de *boosting* para combinar modelos simples y menos precisos en modelos que mejoren la precisión de los casos que no se han predicho correctamente hasta ese momento. El proceso de ajuste de cada árbol se realiza utilizando la técnica de gradiente descendente estocástico (GDE), que es un método de optimización utilizado en el aprendizaje profundo.

2.3. Proceso de optimización de hiperparámetros

Los mecanismos de optimización de hiperparámetros que cuentan con cuatro elementos principales, según lo establece [3]: i) un regresor o un clasificador, ii) un espacio de búsqueda o configuración de los hiperparámetros, iii) el método de optimización para buscar el mejor modelo y iv) función objetivo (métrica de evaluación) para medir y comparar el rendimiento de las diferentes configuraciones de hiperparámetros. Para problemas de OH, el objetivo de este trabajo es obtener:

$$x^* = \arg \max_{x \in X} f(x). \quad (6)$$

x^* es la configuración de hiperparámetros que produce el valor óptimo de $f(x)$; y x es un hiperparámetro que puede tomar cualquier valor en el espacio de búsqueda X . La función objetivo $f(x)$ determina lo bien que se comporta un modelo de aprendizaje.

Esta función compara las predicciones del modelo con la clase real para determinar el nivel de error del algoritmo.

Tabla 1 Hiperparámetros de los algoritmos RF, SVM, XGBoost.

Algoritmo	Hiperparámetro	Descripción
RF	<i>n_estimators</i>	Número de árboles considerados para el modelado
	<i>criterion</i>	Mecanismo para medir la calidad de una división
	<i>max_features</i>	Número de características que se utilizan para construir cada árbol de decisión en el bosque
	<i>max_depth</i>	Profundidad máxima del árbol
	<i>min_samples_split</i>	Número mínimo de muestras que se requieren para dividir un nodo interno en dos subnodos
	<i>min_samples_leaf</i>	Número mínimo de muestras para formar una hoja
SVM	<i>c</i>	Margen entre los vectores de soporte y el hiperplano. Un valor de <i>C</i> bajo permitirá errores en la clasificación, por otro lado, un valor de <i>C</i> más alto buscará un margen de separación más robusto
	<i>kernel</i>	Función matemática que mapea los datos de entrada a un espacio de características de mayor dimensión (<i>linear</i> , <i>rbf</i> , <i>poly</i> , <i>sigmoid</i>)
	<i>degree</i>	Se utiliza con el <i>kernel poly</i> . El grado determina la complejidad de la función de decisión y, por lo tanto, la capacidad del modelo para ajustarse a los datos
XGBoost	<i>n_estimators</i>	Número de árboles a utilizar
	<i>learning_rate</i>	Controla la velocidad de aprendizaje del modelo
	<i>gamma</i>	Controla la complejidad del modelo y evitar el sobreajuste. El valor óptimo de <i>gamma</i> depende de los datos
	<i>subsample</i>	Fracción de observaciones que deben ser muestras aleatorias para cada árbol
	<i>max_depth</i>	Profundidad máxima de los árboles
	<i>colsample_bytree</i>	Fracción de características que se seleccionan al azar en cada árbol

2.3.1. Algoritmos de optimización de hiperparámetros

Los métodos de optimización evalúan de forma eficientes las combinaciones de valores de los hiperparámetros asociados a un algoritmo de aprendizaje automático. Esto permite a los usuarios no expertos desarrollar modelos bien estructurados [14].

La técnica GS [2] funciona evaluando el producto cartesiano de un conjunto finito de valores del espacio de búsqueda definido por el usuario. Es fácil ejecutar GS en paralelo porque cada ensayo se ejecuta individualmente y el resultado es independiente de los de otros ensayos.

Sin embargo, GS sufre la maldición de la dimensionalidad porque el consumo de recursos informáticos aumenta exponencialmente cuando hay hiperparámetros.

Además, un rango de muestreo limitado es aceptable para GS porque no es deseable que haya demasiadas configuraciones.

Por otra parte, RS [3] en lugar de probar todos los valores en el espacio de búsqueda, como es el caso de GS, RS toma aleatoriamente un número predefinido de valores de los límites superior e inferior del espacio de búsqueda como valores de hiperparámetros candidatos, y luego entrena el algoritmo hasta agotar el presupuesto definido. La base

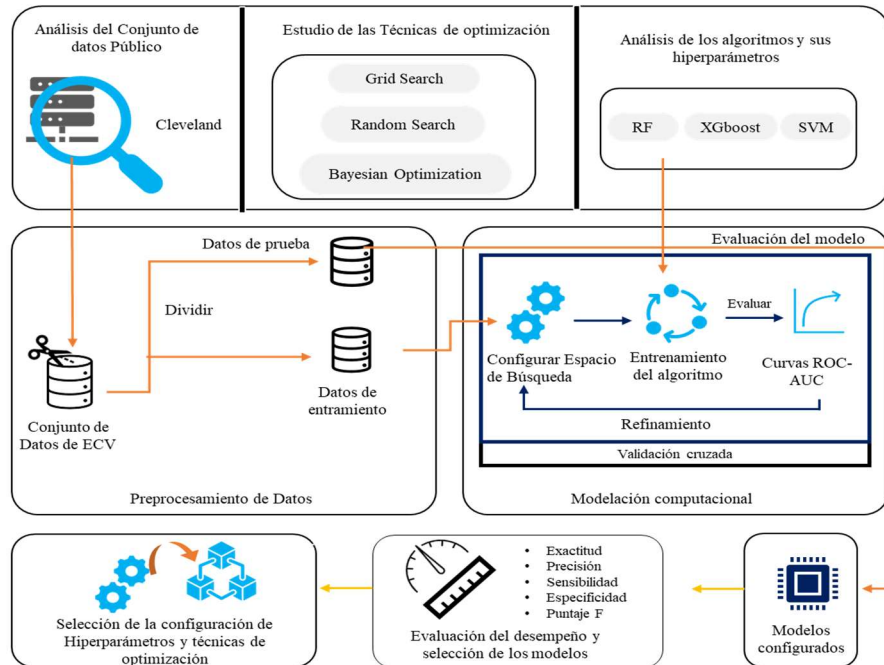


Fig. 1. Metodología de ajuste de Opti.

teórica de RS es que, si el espacio de configuración es lo suficientemente grande, posiblemente se detecte la combinación de hiperparámetros con un óptimo global.

BO [15] utiliza la teoría de la probabilidad bayesiana para encontrar los mejores hiperparámetros de manera más eficiente. BO comienza con la definición de una función objetivo que se desea optimizar, en este caso la función de probabilidad del modelo. A medida que se ejecutan los experimentos, se actualiza la distribución de probabilidad posterior para reflejar los resultados observados. El objetivo es encontrar el conjunto de hiperparámetros que maximice la función objetivo dada la distribución de probabilidad posterior [16].

2.4. Trabajo relacionado

En [17] mostraron que la configuración de los hiperparámetros es a menudo más importante que la elección del algoritmo de aprendizaje automático. Además, en [18] estudiaron la importancia de ajustar determinados hiperparámetros, más que la importancia de afinarlos en su totalidad.

Trabajos como: [3, 19, 21] formularon el proceso de configuración de los hiperparámetros como un problema de optimización, para encontrar los valores de los hiperparámetros que maximicen la tasa de predicciones positivas con respecto a las métricas que evalúan el rendimiento de algoritmos.

La selección de la función objetivo es el primer paso en los métodos de aprendizaje automático. Con la función objetivo determinada, se suelen utilizar métodos exactos y heurísticos para resolver el problema [22].

Tabla 2. Descripción del conjunto de datos *Cleveland*.

Variables	Descripción	Rango
Age	Edad del paciente en años	29-77
Sex	Sexo del paciente	0: hombre, 1: mujer
Cp	Dolor torácico	1: AT, 2: AA, 3: DNA, 4: A
Trestbps	Prensión arterial en reposo	94-200
Chol	Colesterol sérico	126-564
Fbs	Azúcar en sangre en ayunas > 120 mg/dl	0: No, 1: Si
Restecg	Resultados del electrocardiograma	0: normal, 1: anomalía-onda ST-T, 2: hipertrofia del LV
Thalach	Frecuencia cardíaca alcanzada	71-202
Exang	Angina inducida por el ejercicio	0: No, 1: Si
Oldpeak	Depresión del segmento ST	0.0-62.0
Slope	Pendiente del segmento ST de ejercicio máximo	1: pendiente-ascendente, 2: grasa, 3: pendiente-descendente
Ca	Número de vasos mayores coloreados por fuoroscopia	0-3
Thalassemia	Trastorno sanguíneo hereditario que hace que tu cuerpo tenga menos hemoglobina de lo normal	3: normal, 6: defecto-fijo, 7: defecto irreversible
HeartDisease	Variable objetivo	0: ausencia d, 1: presencia

En [23] se optimizaron dos algoritmos SVM para la predicción de EC. El primer modelo es lineal y fue regularizado con la técnica *L1*, mientras que el segundo fue regularizado con la técnica *L2* y utilizaron diferentes *kernels*, incluyendo el lineal y el de la función de base radial (rbf).

Por otro lado, en [24] se estableció un sistema optimizado basado en el algoritmo *XGBoost* para la predicción de EC. Para obtener la solución óptima de forma sistemática y eficaz se aplicó la técnica BO para configurar los hiperparámetros al algoritmo. En [25] se propuso un marco de trabajo basado principalmente en los algoritmos k-Nearest Neighbors (kNN), SVM y RF entrenados con el conjunto de datos de EC *Cleveland*.

3. Predicción de enfermedades cardiovasculares

En esta sección se muestra la metodología de OH aplicada (ver la Fig. 1). Se analiza el conjunto de datos *Cleveland*, las técnicas de optimización GS, RS y BO, y se evalúan los algoritmos de aprendizaje automático RF, SVM y XGBoost para identificar los hiperparámetros más relevantes (ver Tabla 1) con el objeto de construir un modelo de aprendizaje para apoyo en la detección de EC. Posteriormente, se lleva a cabo el

Tabla 3. Estadísticos de las variables cuantitativas del conjunto de datos *Cleveland*.

Variables	Min	Max	Media	Mediana	Moda	Desv. Est.	Varianza
<i>Age</i>	29	77	54.43	56	58	9.039	81.69
<i>Trestbps</i>	94	200	131.68	130	120	17.60	309.75
<i>Chol</i>	126	564	246.69	241	197	51.77	2680.84
<i>Thalach</i>	71	202	149.60	153	162	22.87	523.26
<i>Oldpeak</i>	0	6.2	1.039	0.80	0	1.161	1.34

preprocesamiento del conjunto de datos, que incluye el dividirlo en el conjunto de entrenamiento y prueba. La intención de contar con registros de prueba es para evaluar a los modelos con registros que desconoce y medir el rendimiento a partir de las distintas métricas.

La fase de entrenamiento de cada algoritmo se realizó con un proceso de validación cruzada de 10 particiones. El proceso de modelado computacional se presenta como un proceso iterativo debido a que la búsqueda de los valores óptimos para los hiperparámetros de un algoritmo de aprendizaje puede requerir múltiples iteraciones.

Durante este proceso, se entrenan los algoritmos con diferentes combinaciones de valores de hiperparámetros, y para evaluar el rendimiento del modelo se analiza su curva ROC para determinar su capacidad para distinguir entre las diferentes clases de datos.

Este proceso se repite hasta encontrar una combinación de hiperparámetros que produzca el mejor rendimiento posible en el conjunto de datos dado. Finalmente, se aplican las métricas *Accuracy*, *Precision*, *Recall*, *Specificity* y *F-Score* para seleccionar los modelos con las combinaciones de hiperparámetros que proporcionan mayor rendimiento.

3.1 Descripción del conjunto de datos

El conjunto de datos fue recopilado por el *Cleveland Heart Institute*. Contiene información (demográfica, antecedentes médicos, signos vitales) de 303 personas a los que se le aplicaron una serie de pruebas de diagnóstico; de ellos, 164 presentan EC, el resto (139) tienen ausencia de EC.

En relación con el sexo de los pacientes, el 32% son pacientes mujeres y 68% pacientes hombres. En la Tabla 2 se describen los 14 atributos que contiene el conjunto de datos *Cleveland* y el intervalo de valores que presenta cada variable.

3.2 Análisis de correlación

La correlación se define como una medida estadística que indica la fuerza y la dirección de la relación lineal entre dos variables [26]. Conocer la correlación entre variables es importante porque puede proporcionar información valiosa sobre la relación entre dos o más variables, Para el conjunto de datos *Cleveland* se aplicó la

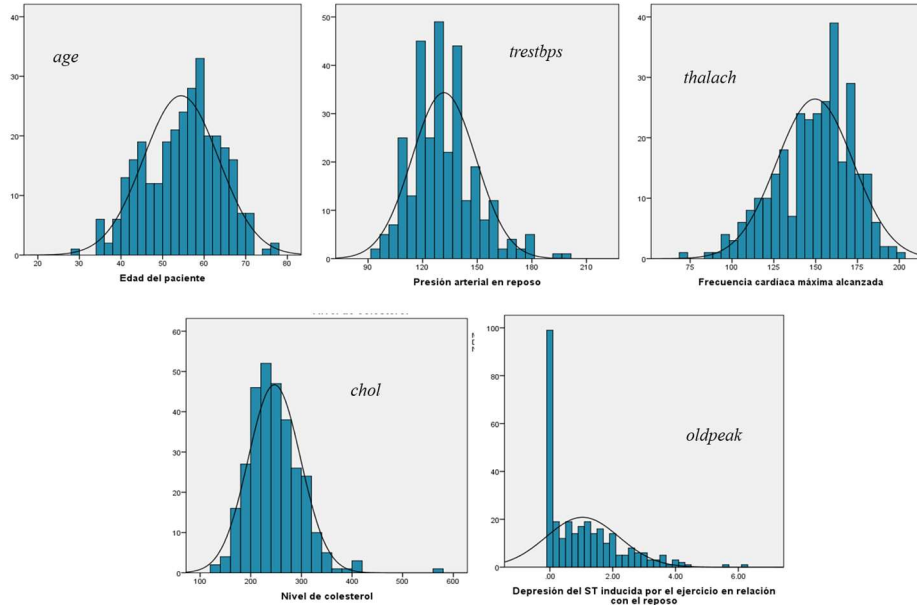


Fig. 2. Histogramas y curva de normalidad de las variables cuantitativas.

correlación de *Spearman* ya que esta técnica es menos sensible a los valores atípicos presentes y es más adecuada para variables que no presentan normalidad en su distribución.

Los pares de variables que presentan mayor relación son: *oldpeak* - *thalach*, *thalach* - *age* y *trestbps* - *age*. Las dos primeras correlaciones son negativas y débiles, lo que indica que a medida que una variable disminuye, la otra variable también. Por otro lado, la última relación es positiva débil, lo que nos hace inferir que a medida que aumenta la edad en las personas también su tensión arterial incrementa.

3.3 Preprocesamiento de datos

El análisis exploratorio de los datos incluye la visualización de los datos a través de histogramas y diagramas de caja y bigotes. El análisis exploratorio de datos es una parte importante del proceso de preprocesamiento de datos porque ayuda a garantizar que los datos estén limpios, sean coherentes y estén listos para su uso en el análisis posterior [27]. En la Tabla 3 se describen los estadísticos descriptivos de las variables cuantitativas del conjunto de datos *Cleveland*.

AT: Angina Típica, AA: Angina Atípica, DNA: Dolor no Anginoso, A: Asintomático

En la Fig. 2 se presentan los histogramas de frecuencias de las variables cuantitativas. Todas las variables presentan un pequeño sesgo en su distribución, por ejemplo, la variable *chol*, *oldpeak* y *trestbps* tienen un sesgo a la izquierda, es decir, tienen una cola asimétrica extendida hacia los valores positivos. Además, conforme el

Tabla 4. Matriz de confusión de los algoritmos RF, SVM y XGBoost.

Técnica	Iter.	Algoritmo	VP	FP	FN	VN	VP+VN
Configuración predeterminada	1	RF	35	7	8	41	76
	1	SVM	37	7	6	41	78
	1	XGBoost	37	9	6	39	76
GS	Todas	RF	36	8	7	40	76
	Todas	SVM	37	7	6	41	78
	Todas	XGBoost	37	9	6	39	76
RS	100	RF	36	5	7	43	79
	100	SVM	34	4	9	44	78
	100	XGBoost	36	7	7	41	77
BO	30	RF	36	4	7	44	80
	30	SVM	37	5	6	43	80
	30	XGBoost	37	6	7	41	78

polígono de frecuencia que se muestra en cada histograma podemos inferir que las distribuciones de cada variable no provienen totalmente a una distribución normal.

Otro tema muy importante en el análisis exploratorio de los datos es poder identificar los datos que se encuentran fuera del rango normal o esperado de los datos, llamados datos atípicos (*outliers*). Las variables con *outliers* son: *Trestbps*, *Chol*, *Thalach* y *Oldpeak*.

Para tratar este tipo de datos se consideró un mecanismo que los reemplaza por los valores más cercanos dentro de un rango definido. Para este caso, en cada atributo se consideró como *outliers* el 1% como extremo inferior y el 3% extremo superior de la distribución.

Existen numerosos algoritmos de aprendizaje automático que necesitan que tanto las variables de entrada como de salida se expresen en términos numéricos, ya que no pueden procesar directamente los datos de las etiquetas en su forma original [28]. El conjunto de datos *Cleveland* presenta variables categóricas las cuales fueron transformadas a partir del método *OneHot Encoding* [24].

El método consiste en crear una columna para cada valor único en la variable categórica original y asignar un valor binario de 0 o 1 a cada columna, según la pertenencia del registro a esa categoría. Las variables que se procesaron mediante la técnica fueron: *Cp*, *Restecg*, *Slope*, *Ca* y *Thalassemia*.

Además, cuando las variables tienen distintas escalas, es importante tener en cuenta que esto puede afectar la interpretación de los análisis y los modelos estadísticos. Es posible que las variables con mayores escalas tengan una influencia desproporcionada en los resultados en comparación con las variables con escalas más pequeñas.

Se ha identificado que los datos de dominios médicos son en su mayoría discretos, por lo que estandarizarlos es esencial para hacer converger las características de estos. En este trabajo se consideró un mecanismo de estandarización robusta para generar una nueva escala. Es uno de los métodos más populares para definir una única escala para todas las características utilizando los percentiles 25 y 75 de la distribución de datos para cada característica.

Tabla 5. Métricas de desempeño de los algoritmos RF, SVM y XGBoost.

Técnica.	Iter.	Algoritmo	Acc. Train	Acc. Test	Precision	Recall	Specifity	F-score
Configuración predeterminada	1	RF	81.14	83.52	83.33	81.40	85.42	82.35
	1	SVM	84.50	85.71	84.09	86.05	85.42	85.06
	1	XGBoost	80.62	83.52	80.43	86.05	81.25	83.15
GS	Todas	RF	84.91	83.52	81.82	83.72	83.33	82.76
	Todas	SVM	84.91	85.71	84.09	86.05	85.42	85.06
	Todas	XGBoost	84.43	83.52	80.43	86.05	81.25	83.15
RS	100	RF	87.99	86.81	87.80	83.72	89.58	85.71
	100	SVM	88.39	85.71	89.47	79.07	91.67	83.95
	100	XGBoost	88.78	84.62	83.72	83.72	85.42	83.72
BO	30	RF	83.93	87.91	90.00	83.72	91.67	86.75
	30	SVM	83.09	87.91	88.10	86.05	89.58	87.06
	30	XGBoost	84.02	85.71	86.05	84.09	87.23	85.06

4. Resultados y discusión

En esta sección se exponen los resultados tras aplicar las técnicas de optimización. Los experimentos se llevaron a cabo mediante el uso de diversas bibliotecas de aprendizaje automático, tales como *scikit-learn* y *optunity*, junto con el lenguaje de programación *Python*. Asimismo, cabe destacar que la experimentación fue realizada en una computadora HP-Victus con un procesador Core™ i5-12450H de 2.00 GHz, 8 GB de memoria RAM y con sistema operativo Windows 11.

Como primer paso, se procedió a dividir el conjunto de datos *Cleveland* en dos subconjuntos. El 70% de las muestras (aproximadamente 212 registros) se utilizaron para la fase de entrenamiento de los tres algoritmos, utilizando una validación cruzada de 10 *foldds*. Además, se definió un número máximo de iteraciones o combinaciones que se evaluarán por parte de las técnicas RS y BO.

Finalmente, se empleó el 30% de las muestras (cerca de 91 registros) para la etapa de prueba de los modelos previamente entrenados. En la Tabla 4 se muestra el total de registros clasificados como: verdaderos positivos (VP), verdaderos negativos (VN), falsos positivos (FP) y falsos negativos (FN) de los modelos que mostraron mayor rendimiento en relación con las técnicas de tres técnicas de OH. Además, se muestra el total de registros clasificados correctamente.

En la Tabla 5 se presentan los resultados de los modelos al evaluar su desempeño mediante las métricas *Accuracy*, *Precision*, *Recall*, *Specificity* y *F-Score*, evaluados con el conjunto de prueba.

Se observa que la técnica de optimización BO genera los modelos RF, SVM y XGBoost que presentan mayor desempeño, en relación con la métrica *Accuracy*. En términos generales estos modelos han logrado predecir correctamente la mayoría de las muestras del conjunto de datos de prueba en relación con el conjunto total de muestras de prueba.

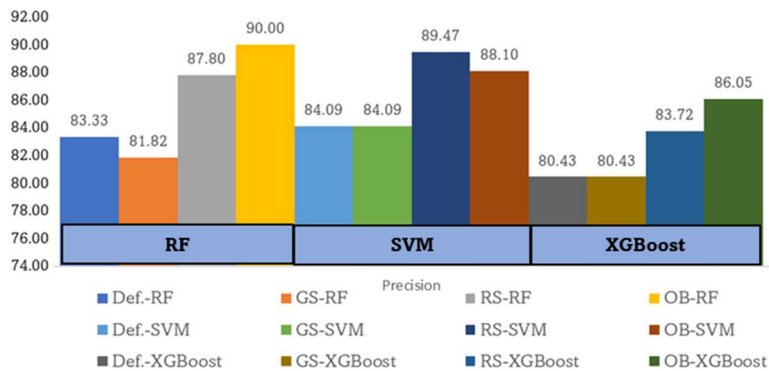


Fig. 3. Comparación de la métrica *precision* de modelos resultantes.

Para identificar el modelo con mayor capacidad para predecir a las personas que padecen EC, es importante enfocarnos en la métrica de *precision*. En la Fig. 3 se puede apreciar que en términos generales los tres modelos (RF, SVM, XGBoost) presentaron mayor rendimiento en diversos procedimientos de ajuste de hiperparámetros con respecto a la configuración de los hiperparámetros por defecto. Utilizando la técnica de optimización BO el modelo RF alcanzó el mayor rendimiento (90%).

4.1 Configuraciones de hiperparámetros identificadas con mejor desempeño

La Tabla 6 muestra para cada algoritmo las configuraciones de hiperparámetros que se obtuvieron a partir de las técnicas de OH. Estas configuraciones dan al modelo el mayor desempeño en relación con el espacio de búsqueda propuesto. Se observa que algunos valores de los hiperparámetros varían en cada técnica de OH.

Se ha realizado un comparativo entre los resultados de los modelos propuesto en este trabajo con algunos modelos reportados en la literatura. El desempeño de los modelos configurados mediante la técnica BO supera a los modelos propuestos en [27]. Además, el rendimiento obtenido por los modelos RF-RS y RF-BO es superior al de [28], donde el modelo RF obtiene un 83% en la métrica *Accuracy* en la fase de entrenamiento.

5. Conclusiones y trabajo a futuro

En este trabajo se construyeron modelos de predicción de EC a partir de un enfoque experimental que optimiza los hiperparámetros y mejora el desempeño de los modelos. Como parte del proceso se describieron los hiperparámetros más importantes de los algoritmos RF, SVM y XGBoost que tienen un impacto directo en la fase de entrenamiento. Así mismo planteamos el enfoque de OH y describimos las características de las técnicas de optimización GS, RS y BO.

El principal reto de este trabajo fue definir los mecanismos de entrenamiento para los tres algoritmos de aprendizaje automático mediante las técnicas de OH. Además, se consideró un mecanismo adicional que presenta a los modelos con la configuración predeterminada de sus hiperparámetros. Este mecanismo sirvió de referencia para el hacer el contraste de la precisión de cada modelo cuando se entrenan a partir de técnicas

Tabla 6. Configuraciones de hiperparámetros con mejor desempeño.

Algoritmo	Hiperparámetro	GS	RS	BO
RF	<i>n_estimators</i>	50	52	30
	<i>max_features</i>	0.1	1.0	0.2
	<i>max_depth</i>	5	5	9
	<i>min_samples_split</i>	2	7	7
	<i>min_samples_leaf</i>	6	5	4
	<i>bootstrap</i>	False	True	True
SVM	<i>c</i>	2	40	14
	<i>gamma</i>	0.1	0.01	0.001
	<i>kernel</i>	rfb	poly	sigmoid
	<i>degree</i>	1	2	4
XGBoost	<i>n_estimators</i>	45	55	65
	<i>learning_rate</i>	0.1	0.06	0.15
	<i>gamma</i>	1	6.4	0.4
	<i>subsample</i>	0.8	0.5	0.8
	<i>max_depth</i>	5	5	7

de optimización. Como resultado de los experimentos se identificó que los modelos RF y SVM optimizados con las técnicas BO y RS, respectivamente, presentan un mayor desempeño. La búsqueda de los valores de cada hiperparámetro estuvo restringido al espacio de búsqueda que se planteó en cada mecanismo experimental.

En el estado del arte se han planteado métodos de optimización heurísticos, incluyendo técnicas basadas en inteligencia de enjambre y algoritmos evolutivos. Como trabajo a futuro se plantearán el desarrollo de mecanismos de configuración de hiperparámetros considerando estas técnicas.

Referencias

1. Probst, P., Bischl, B., Boulesteix, A.: Tunability: importance of hyperparameters of machine learning algorithms. *The Journal of Machine Learning Research*, vol. 20, no. 1, pp 1934–1965 (2018) doi: 10.48550/ARXIV.1802.09596
2. Yu, T., Zhu, H.: Hyper-parameter optimization: a review of algorithms and applications (2020) doi: 10.48550/ARXIV.2003.05689
3. Yang, L., Shami, A.: On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, vol. 415, pp. 295–316 (2020) doi: 10.1016/j.neucom.2020.07.061
4. Thornton, C., Hutter, F., Hoos, H. H., Leyton-Brown, K.: Auto-weka. In: *Proceedings of the 19th ACM International Conference on Knowledge Discovery and Data Mining*, pp. 847–855 (2013) doi: 10.1145/2487575.2487629
5. Detrano, R.: Enfermedades cardiovasculares. Organización Panamericana de la Salud (2023) www.paho.org/es/temas/enfermedades-cardiovasculares
6. Breiman, L.: Random forests. *Machine Learning*, vol. 45, no. 1, pp. 5–32 (2001) doi: 10.1023/a:1010933404324
7. Breiman, L.: Support-vector networks. *Machine Learning*, vol. 45, no. 1, pp. 5–32 (2001) doi: 10.1023/a:1010933404324

8. Chen, T., Guestrin, C.: Xgboost. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016) doi: 10.1145/2939672.2939785
9. Paredes, G. E., Rodríguez, A. V.: Aplicaciones de programación no lineal. *OmniaScience Scholar* (2016) doi: 10.3926/oss.21
10. Gambella, C., Ghaddar, B., Naoum-Sawaya, J.: Optimization problems for machine learning: A survey. *European Journal of Operational Research*, vol. 290, no. 3, pp. 807–828 (2021) doi: 10.1016/j.ejor.2020.08.045
11. Elgeldawi, E., Sayed, A., Galal, A. R., Zaki, A. M.: Hyperparameter tuning for machine learning algorithms used for arabic sentiment analysis. *Informatics*, vol. 8, no. 4, pp. 79 (2021) doi: 10.3390/informatics8040079
12. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, É.: Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830 (2011)
13. Müller, A. C., Guido, S.: Introduction to machine learning with python: A guide for data scientists. O'Reilly Media (2016)
14. Schmidt, M., Safarani, S., Gastinger, J., Jacobs, T., Nicolas, S., Schulke, A.: On the performance of differential evolution for hyperparameter tuning. In: International Joint Conference on Neural Networks, pp. 1–8 (2019) doi: 10.1109/ijenn.2019.8851978
15. Injadat, M., Salo, F., Nassif, A. B., Essex, A., Shami, A.: Bayesian optimization with machine learning algorithms towards anomaly detection. In: IEEE Global Communications Conference, pp. 1–6 (2018) doi: 10.1109/glocom.2018.8647714
16. Hazan, E., Klivans, A., Yuan, Y.: Hyperparameter optimization: A spectral approach. In: 6th International Conference on Learning Representations (2017) doi: 10.48550/ARXIV.1706.00764
17. Lavesson, N., Davidsson, P.: Quantifying the impact of learning algorithm parameter tuning. In: Proceedings of the 21st National Conference on Artificial Intelligence, vol. 1, pp. 395–400 (2006)
18. Weerts, H. J. P., Mueller, A. C., Vanschoren, J.: Importance of tuning hyperparameters of machine learning algorithms. *Journal of Machine Learning Research*, vol. 20, pp. 1–32 (2019) doi: 10.48550/ARXIV.2007.07588
19. Pannakkong, W., Thiwa-Anont, K., Singthong, K., Parthanadee, P., Buddhakulsomsiri, J.: Hyperparameter tuning of machine learning algorithms using response surface methodology: A case study of ANN, SVM and DBN. *Mathematical Problems in Engineering*, vol. 2022, pp. 1–17 (2022) doi: 10.1155/2022/8513719
20. Khourdifi, Y., Bahaj, M.: Heart disease prediction and classification using machine learning algorithms optimized by particle swarm optimization and ant colony optimization. *International Journal of Intelligent Engineering and Systems*, vol. 12, no. 1, pp. 242–252 (2019) doi: 10.22266/ijies2019.0228.24
21. Andonie, R.: Hyperparameter optimization in learning systems. *Journal of Membrane Computing*, vol. 1, no. 4, pp. 279–291 (2019) doi: 10.1007/s41965-019-00023-0
22. Ali, L., Niamat, A., Khan, J. A., Golilarz, N. A., Xingzhong, X., Noor, A., Nour, R., Bukhari, S. A. C.: An optimized stacked support vector machines based expert system for the effective prediction of heart failure. *IEEE Access*, vol. 7, pp. 54007–54014 (2019) doi: 10.1109/access.2019.2909969
23. Budholiya, K., Shrivastava, S. K., Sharma, V.: An optimized XGBoost based diagnostic system for effective prediction of heart disease. *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 7, pp. 4514–4523 (2022) doi: 10.1016/j.jksuci.2020.10.013

24. Gupta, A., Kumar, R., Arora, H. S., Raman, B.: MIFH: A machine intelligence framework for heart disease diagnosis. *IEEE Access*, vol. 8, pp. 14659–14674 (2020) doi: 10.1109/access.2019.2962755
25. El-Hashash, E. F., Shiekh, R. H. A.: A comparison of the Pearson, Spearman rank and Kendall tau correlation coefficients using quantitative variables. *Asian Journal of Probability and Statistics*, pp. 36–48 (2022) doi: 10.9734/ajpas/2022/v20i3425
26. Chang, V., Bhavani, V. R., Xu, A. Q., Hossain, M.: An artificial intelligence model for heart disease detection using machine learning algorithms. *Healthcare Analytics*, vol. 2, pp. 100016 (2022) doi: 10.1016/j.health.2022.100016
27. Rajeswari, D., Thangavel, K.: The performance of data normalization techniques on heart disease datasets. *International Journal of Advanced Research in Engineering and Technology*, vol. 11, no. 12, pp. 2350–2357 (2020) doi: 10.34218/IJARET.11.12.2020.222