

Uso de un método de clasificación de Machine Learning para la elaboración de un modelo como apoyo en el Diagnóstico de Enfermedades Cardíacas Using a Machine Learning classification method to develop a model to support the Diagnosis of Heart Disease

Escalante Vega, J. E.¹, Del Toro González, R. A.², Salem Silva, F. S.²

¹ Facultad de Estadística e informática, Universidad Veracruzana

² Facultad de Matemáticas, Universidad Veracruzana

¹ jescalante@uv.mx, ² zs20016138@estudiantes.uv.mx, frsalem@uv.mx

Fecha de recepción: 20 de septiembre de 2024

Fecha de aceptación: 31 de marzo de 2025

Resumen. El exceso de información en medicina, junto con la publicación constante de estudios con resultados contradictorios, ha hecho que el diagnóstico médico, especialmente en enfermedades cardíacas, sea cada vez más complejo. En este trabajo se aplicó un enfoque cuantitativo mediante técnicas de aprendizaje automático, utilizando el algoritmo de bosques aleatorios implementado en el lenguaje Julia. A partir del “Heart Disease dataset”, se construyó y validó un modelo de clasificación capaz de predecir si una persona tiene probabilidad de padecer una enfermedad cardíaca. El modelo fue entrenado y evaluado mediante validación cruzada y un conjunto de prueba, alcanzando una exactitud del 72%. Estos resultados muestran el potencial del aprendizaje automático como apoyo en la toma de decisiones clínicas.

Palabras Clave: Diagnóstico Médico, Conjunto de Datos, Aprendizaje Automático, Bosques Aleatorios, Julia.

Summary. The excess of information in medicine, together with the constant publication of studies with contradictory results, has made medical diagnosis—especially for heart diseases—increasingly complex. This work applies a quantitative approach using machine learning techniques, specifically the random forest algorithm implemented in the Julia programming language. Based on the “Heart Disease dataset,” a classification model was built and validated to predict whether a person is likely to suffer from heart disease. The model was trained and evaluated through cross-validation and a test dataset, achieving an accuracy of 72%. These results demonstrate the potential of machine learning as a support tool in clinical decision-making.

Keywords: Medical Diagnosis, Database, Machine Learning, Random Forests, Julia.

1 Introducción

Las enfermedades cardíacas representan una de las principales causas de morbilidad y mortalidad a nivel mundial. De acuerdo con datos de la Organización Mundial de la Salud, los trastornos cardiovasculares son responsables de aproximadamente el 30% de las muertes anuales en el mundo [1]. Esta realidad subraya la necesidad urgente de contar con herramientas de diagnóstico más eficientes, que permitan una detección temprana y precisa de factores de riesgo.

Sin embargo, el diagnóstico clínico tradicional enfrenta múltiples desafíos: desde la dependencia de pruebas de laboratorio costosas y especializadas, hasta la dificultad de integrar grandes volúmenes de datos clínicos, a menudo heterogéneos y con relaciones no lineales [1]. Este panorama se complica aún más por la presencia de información contradictoria en la literatura médica, lo que genera incertidumbre en la toma de decisiones clínicas.

En este contexto, los métodos de aprendizaje automático (Machine Learning) se han posicionado como una alternativa prometedora para apoyar el diagnóstico médico. Entre estos, el algoritmo de bosques aleatorios ha demostrado ser especialmente eficaz para tareas de clasificación, gracias a su capacidad de manejar datos con múltiples variables y su resistencia al sobreajuste [2][3].

El objetivo de esta investigación es desarrollar un modelo predictivo utilizando el algoritmo de bosques aleatorios, que permita estimar si una persona tiene probabilidad de padecer una enfermedad cardíaca [4]. Para ello, se utilizará el “Heart Disease dataset” del UCI Machine Learning Repository [5], del cual se seleccionarán 14 atributos clínicos relevantes. La construcción del modelo se llevará a cabo en dos etapas:

- Limpieza de datos: Utilizando los paquetes proporcionados por el lenguaje Julia (por ejemplo, `DataFrame.jl`, `Plots.jl`, etc.), verificaremos la existencia de valores atípicos, valores faltantes, así como

las tendencias y relaciones entre las diferentes características del conjunto de datos para obtener una mejor comprensión de la información disponible y derivar ideas útiles a partir de ella [3] [4] [6].

- Construcción del modelo:
 - Selección del modelo [2][3][7].
 - Entrenamiento del modelo [3][7] [8].
 - Validación del modelo [9][10].

Este trabajo no solo busca construir un modelo con buen desempeño predictivo, sino también explorar el potencial del aprendizaje automático como herramienta de apoyo en contextos clínicos, contribuyendo a mejorar la precisión diagnóstica y la toma de decisiones médicas.

2 Marco teórico

El aprendizaje automático (Machine Learning) ha cobrado gran relevancia en los últimos años debido a su capacidad para automatizar procesos de análisis y predicción a partir de datos. En el contexto médico, estos modelos pueden utilizarse para apoyar el diagnóstico clínico mediante la identificación de patrones complejos en los datos de pacientes. A continuación, se describen las principales etapas que conforman un sistema de clasificación supervisado, con el fin de establecer el marco conceptual que sustenta la presente investigación.

2.1 Definición del Problema

El primer paso en un proyecto de aprendizaje automático consiste en entender con claridad el problema que se desea resolver. En el caso de modelos de clasificación, se trata de predecir una etiqueta o clase a partir de un conjunto de características. El correcto planteamiento del problema permite seleccionar adecuadamente las técnicas de modelado, las métricas de evaluación y los datos relevantes [3][4][8][10].

2.2 Recolección y Preparación de Datos

Los datos son esenciales en los modelos de Machine Learning, y su calidad afecta directamente el desempeño del modelo. Este paso incluye:

- Recolección de datos: Obtener los datos necesarios desde diferentes fuentes, como bases de datos, archivos CSV, etc.
- Limpieza de datos: Eliminar datos erróneos, duplicados o valores faltantes.
- Transformación de datos: Convertir los datos a un formato adecuado para el algoritmo mediante técnicas como normalización o codificación de variables categóricas.
- División de datos: Los datos se dividen en dos conjuntos: el conjunto de entrenamiento y el conjunto de prueba.

¿Por qué dividir los datos?

Dividir los datos es crucial para evitar que el modelo se sobreajuste (memorice los datos del entrenamiento). La propuesta es que el modelo no solo aprenda a realizar predicciones precisas con los datos con los que ha sido entrenado, sino que también tenga la capacidad de generalizar y hacer estas predicciones en conjuntos de datos no analizados previamente. [3][4][6].

2.3 Selección del Modelo (Enfoque en bosques aleatorios)

En este paso se elige el modelo de Machine Learning, en este caso un bosque aleatorio, que mejor se ajusta al problema. Esta técnica se basa en la construcción de múltiples árboles de decisión, donde cada uno es un modelo basado en una estructura de árbol, que consta de:

- Nodos internos: Representan características o variables de decisión en el conjunto de datos.
- Ramas: Representan los resultados de esas decisiones o criterios.
- Hojas: Representan la clasificación o predicción final.

Un árbol de decisión funciona dividiendo iterativamente los datos en subconjuntos más pequeños basándose en una característica que maximice algún criterio, con el objetivo de crear hojas que contengan datos lo más homogéneos posibles. [2][3][9].

2.4 Entrenamiento del Modelo

El entrenamiento es el proceso donde el modelo “aprende” a partir del conjunto de entrenamiento. Este conjunto representa alrededor del 70-80 % de los datos y es donde el modelo ajusta sus parámetros para identificar patrones. Durante el entrenamiento:

- El modelo ajusta sus parámetros internos con el objetivo de minimizar el error entre las predicciones y los resultados reales.
- En el caso del árbol de decisión, el modelo construye una estructura de árbol donde cada nodo representa una decisión basada en una característica de los datos.
- El modelo sigue dividiendo los datos en ramas hasta que cada nodo final (hoja) contenga datos que sean lo más homogéneos posible respecto a la variable objetivo.

Conjunto de Entrenamiento: Se usa para entrenar el modelo, es decir, para que el algoritmo ajuste sus parámetros y aprenda los patrones en los datos. Aquí el modelo optimiza su desempeño buscando minimizar el error de predicción en este conjunto.

Conjunto de Prueba: No se usa durante el entrenamiento. Sirve para evaluar el rendimiento del modelo una vez que ha sido entrenado, con el fin de comprobar si ha aprendido bien los patrones y si puede generalizar a nuevos datos. Representa el 20-30 % de los datos y es fundamental para obtener una medida objetiva de la capacidad del modelo. [3][7][8].

2.5 Evaluación del Modelo

Después de entrenar el modelo, se necesita evaluar su rendimiento utilizando el conjunto de prueba. Este conjunto no fue usado durante el entrenamiento, por lo que permite ver cómo el modelo se comporta con datos nuevos, simulando su desempeño en la realidad.

Esta evaluación garantiza que el modelo no se sobreajuste a los datos de entrenamiento y que, en su lugar, aprenda patrones que puedan generalizarse. [9][10].

2.6 Optimización del Modelo

Si el modelo no ofrece un rendimiento adecuado, puede ser necesario ajustarlo. En el caso de un árbol de decisión, se puede optimizar mediante:

- Poda: Reducción del tamaño del árbol eliminando ramas que no agregan valor significativo. [7][9]
- Ajuste de hiperparámetros: Ajustar la profundidad máxima del árbol o el número mínimo de ejemplos en una hoja. [8].

2.7 Predicción y Despliegue

Una vez optimizado y validado el modelo, está listo para hacer predicciones sobre nuevos datos. El modelo se utiliza en producción para realizar tareas automatizadas de predicción. [10][11].

3 Antecedentes

En los últimos años, el uso de algoritmos de aprendizaje automático ha tomado relevancia en el ámbito médico debido a su capacidad para procesar grandes volúmenes de datos clínicos y detectar patrones complejos que los métodos tradicionales no logran identificar [5][7][10]. Particularmente en el diagnóstico de enfermedades cardíacas, se han desarrollado múltiples estudios que exploran modelos predictivos basados en técnicas supervisadas como máquinas de vectores de soporte, redes neuronales y árboles de decisión [9][11].

Entre estas técnicas, los bosques aleatorios (Random Forest) han mostrado ventajas importantes, como su robustez frente a datos ruidosos, su capacidad para evitar el sobreajuste y su habilidad para manejar relaciones no lineales entre variables [2][6]. Investigaciones como las de Matloff (2024) y Geetha & Sendhilkumar (2023) destacan que los modelos de bosques aleatorios alcanzan niveles de precisión comparables o superiores a los de métodos más complejos, sin sacrificar interoperabilidad, lo cual es una ventaja clave en el ámbito clínico [7][10].

Además, autores como Zhou (2021) y Smolyakov (2023) plantean que este tipo de modelos no solo ofrecen buenas métricas de rendimiento, sino que también permiten extraer la importancia de las variables en el proceso de predicción, facilitando así la validación por parte de expertos médicos [9][11].

A pesar de estos avances, muchos estudios previos se han centrado en conjuntos de datos limitados o en implementaciones genéricas de estos algoritmos, sin un enfoque específico en la validación práctica del modelo en contextos reales de toma de decisiones. Por tanto, esta investigación propone como innovación el desarrollo de un modelo predictivo utilizando bosques aleatorios implementados en el lenguaje Julia, aplicado específicamente al conjunto de datos “Heart Disease” del UCI Repository [8]. La combinación de esta técnica, el lenguaje de programación orientado a rendimiento y un enfoque en características clínicas seleccionadas pretende aportar un modelo más eficiente y transparente para su futura aplicación como herramienta de apoyo diagnóstico.

4 Desarrollo del modelo

4.1 Recolección y preparación de datos

El pre procesamiento de datos es una etapa crucial en cualquier proyecto de aprendizaje automático, ya que permite preparar la información de manera adecuada para su análisis [4]. Antes de construir un modelo de clasificación, es necesario garantizar que el conjunto de datos esté en un formato utilizable, sin valores faltantes y con las características relevantes correctamente estructuradas.

Las características de la base de datos son las siguientes:

- Age: Edad del paciente.
- Sex: Sexo del paciente, donde 0 representa femenino y 1 representa masculino.
- Angina es un tipo de dolor de pecho que ocurre cuando el corazón no recibe sangre que contenga suficiente oxígeno.
- Cp: Tipo de dolor de pecho, donde las categorías son las siguientes:

Tabla 1 Tipos de dolor de pecho.

Categoría	Tipo de dolor de pecho
0	Angina típica
1	Angina atípica
2	Dolor no anginoso
3	Asintomático (no presenta síntomas de enfermedad)

- Trestbps: Presión arterial en reposo en mm Hg al momento de la admisión al hospital.
- Chol: Colesterol sérico en mg/dl.
- Fbs: Glucemia en ayunas, indicando si es mayor a 120 mg/dl o no, donde 0 representa falso y 1 representa verdadero.
- Restecg: Resultados de ECG en reposo, donde las categorías son las siguientes:

Tabla 2 Resultados del ECG.

Categoría	Resultados
0	Normal
1	Anomalía de la onda ST-T (inversión de la onda T y/o elevación o depresión del ST de > 0.05 mV)
2	Muestra hipertrofia ventricular izquierda probable o definitiva según los criterios de Estes

- Thalach: Máxima frecuencia cardíaca registrada.

- Exang: Presencia de angina inducida por el ejercicio, donde 0 representa no y 1 representa sí.
- Oldpeak: Depresión del segmento ST inducida por el ejercicio en relación al estado de reposo.
- La depresión del ST es más severa y extendida cuanto más profunda sea, y cuanto mayor sea la extensión de la depresión ST, más grave es la enfermedad. La depresión del ST se refiere a una depresión que se encuentra en un electrocardiograma. Esto ocurre cuando la traza del segmento ST está inusualmente por debajo de la línea base normal. El segmento ST se refiere al final de la onda S y el comienzo de la onda T en un electrocardiograma.
- Slope: La pendiente del segmento ST durante el ejercicio máximo. Las categorías son las siguientes:

Tabla 3 Pendiente del pico del ejercicio.

Categoría	Pendiente del pico
0	Pendiente ascendente
1	Plana
2	Pendiente descendente

- Ca: Número de vasos principales coloreados por fluoroscopia.
- Thal: Talasemia.
- La talasemia es una enfermedad hereditaria de la sangre en la cual el cuerpo no puede producir una forma adecuada o apropiada de hemoglobina.
- Target: Diagnóstico de enfermedad cardíaca, donde 0 representa que el estrechamiento del diámetro es menor al 50% y 1 representa que el estrechamiento del diámetro es mayor al 50%.

El primer paso será cargar los datos y verificar su estructura inicial. A continuación, se presenta el código para importar el conjunto de datos y realizar las primeras inspecciones:

Algoritmo 1.

```
[3]: using CSV
using HTTP
using DataFrames
url3="https://raw.githubusercontent.com/PacktWorkshops/The-Data-Analysis-Workshop/master/Chapter07/Dataset/heart.csv"
res3 = HTTP.get(url3)
medical = DataFrame(CSV.File(res3.body))
first(medical,5)
```

[3]: 5 rows × 14 columns (omitted printing of 3 columns)

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope
	Int64	Int64	Int64	Int64	Int64	Int64	Int64	Int64	Int64	Float64	Int64
1	63	1	3	145	233	1	0	150	0	2.3	0
2	37	1	2	130	250	0	1	187	0	3.5	0
3	41	0	1	130	204	0	0	172	0	1.4	2
4	56	1	1	120	236	0	1	178	0	0.8	2
5	57	0	0	120	354	0	1	163	1	0.6	2

A continuación, obtenemos las estadísticas descriptivas de la base.

```
[4]: using Distributions
describe(medical)
```

[4]: 14 rows × 7 columns

	variable	mean	min	median	max	nmissing	eltype
	Symbol	Float64	Real	Float64	Real	Int64	DataType
1	age	54.3663	29	55.0	77	0	Int64
2	sex	0.683168	0	1.0	1	0	Int64
3	cp	0.966997	0	1.0	3	0	Int64
4	trestbps	131.624	94	130.0	200	0	Int64
5	chol	246.264	126	240.0	564	0	Int64
6	fbs	0.148515	0	0.0	1	0	Int64
7	restecg	0.528053	0	1.0	2	0	Int64
8	thalach	149.647	71	153.0	202	0	Int64
9	exang	0.326733	0	0.0	1	0	Int64
10	oldpeak	1.0396	0.0	0.8	6.2	0	Float64
11	slope	1.39934	0	1.0	2	0	Int64
12	ca	0.729373	0	0.0	4	0	Int64
13	thal	2.31353	0	2.0	3	0	Int64
14	target	0.544554	0	1.0	1	0	Int64

Algoritmo 2.

Este DataFrame cuenta con 14 columnas, todas de tipo numérico. Cada columna contiene 303 registros, lo que indica que el conjunto de datos está compuesto por información de salud de 303 pacientes y no contiene valores faltantes. Dado que algunos nombres de las columnas pueden resultar confusos para quienes no tienen formación en el área médica, es conveniente renombrar algunas para facilitar su comprensión.

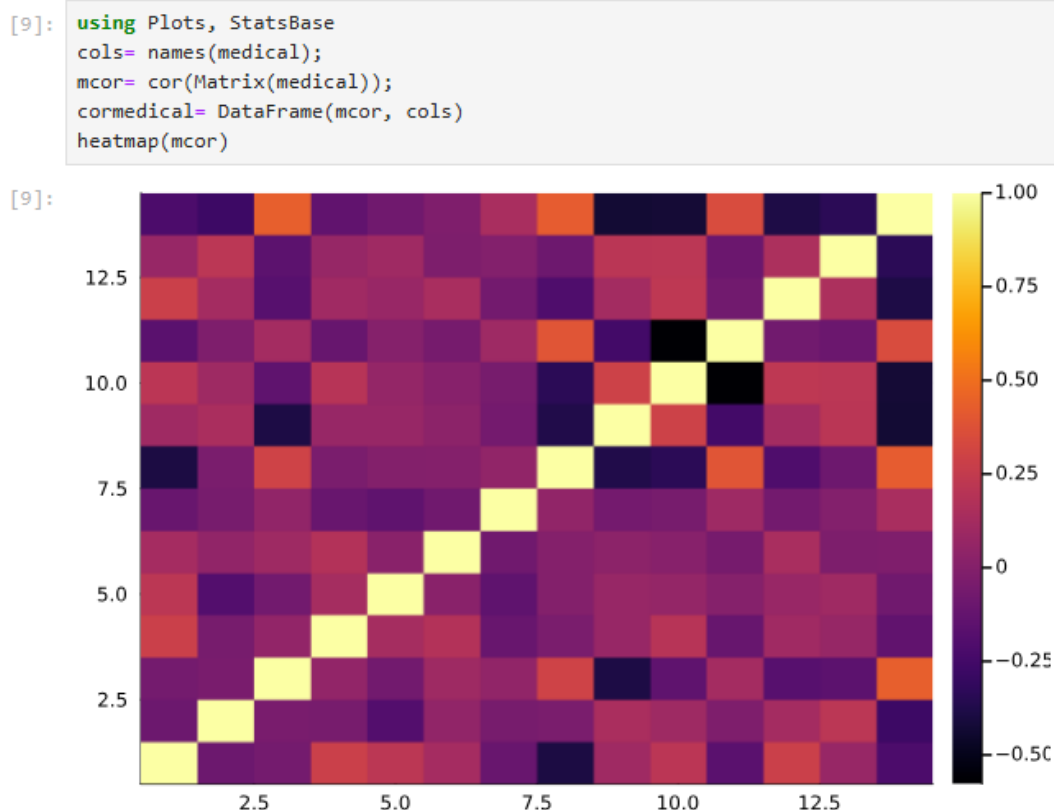
```
[12]: rename!(medical, Dict("cp" => "chest_pain",
                           "trestbps" => "rest_bp",
                           "fbs" => "fast_bld_sugar",
                           "restecg" => "rest_ecg",
                           "thalach" => "max_hr",
                           "exang" => "ex_angina",
                           "oldpeak" => "st_depr",
                           "ca" => "colored_vessels",
                           "thal" => "thalassemia" ))
first(medical,1)
```

[12]: 1 rows × 14 columns (omitted printing of 5 columns)

	age	sex	chest_pain	rest_bp	chol	fast_bld_sugar	rest_ecg	max_hr	ex_angina
	Int64	Int64	Int64	Int64	Int64	Int64	Int64	Int64	Int64
1	63	1	3	145	233	1	0	150	0

Algoritmo 3.

En este trabajo, se construirá un modelo de aprendizaje automático, por lo que es fundamental identificar cuáles de estas características realmente afectan a la variable objetivo. Estas serán las características que se incorporarán al modelo para garantizar que este aprenda los patrones y tendencias adecuados, evitando así incluir información que no influya en las predicciones.



Algoritmo 4.

El mapa de calor anterior indica que solo unas pocas características, como la pendiente, la frecuencia cardíaca máxima y el tipo de dolor en el pecho, presentan una correlación positiva significativa con la columna objetivo.

4.2 Entrenamiento del modelo

Se abordará la construcción de un modelo de clasificación para el diagnóstico de enfermedades cardíacas. La importancia de este tipo de modelos radica en su capacidad para analizar grandes volúmenes de datos y detectar patrones que permiten predecir con precisión la probabilidad de que un paciente presente una enfermedad [2]. El proceso de modelado se llevará a cabo mediante la implementación de algoritmos de clasificación, utilizando bosques aleatorios, que han demostrado ser efectivos en problemas similares [8].

Un paso crucial en la construcción de modelos de clasificación es la separación de los datos en dos conjuntos: datos de entrenamiento y datos de prueba. La proporción de 80/20 se seleccionó para asegurar que el modelo disponga de suficiente información para entrenarse adecuadamente y, a la vez, cuente con suficientes instancias para una evaluación significativa.

El modelo seleccionado para la clasificación es un bosque aleatorio (*Random Forest*). Esta técnica se basa en la construcción de múltiples árboles de decisión, donde cada uno se entrena con un subconjunto aleatorio de datos y características. Luego, los resultados de estos árboles se combinan para mejorar la precisión de la predicción y reducir el riesgo de sobreajuste [2]. El uso de bosques aleatorios es especialmente útil en problemas donde los datos presentan mucha variabilidad o cuando se busca minimizar el sesgo inherente a los modelos individuales [7]. En este caso, aplicamos el bosque aleatorio para analizar la presencia de enfermedades cardíacas, ya que permite capturar patrones complejos en los datos y obtener predicciones más robustas.

Algoritmo 5.

```
[6]: using DecisionTree

# Crear el modelo
forest_model = RandomForestClassifier()

# Entrenar el modelo
DecisionTree.fit!(forest_model, X_train_matrix, y_train_vector)

[6]: RandomForestClassifier
n_trees:          10
n_subfeatures:    -1
partial_sampling: 0.7
max_depth:        -1
min_samples_leaf: 1
min_samples_split: 2
min_purity_increase: 0.0
classes:          [0, 1]
ensemble:         Ensemble of Decision Trees
Trees:            10
Avg Leaves:       32.8
Avg Depth:        9.2
```

El modelo se configuró con **10 árboles de decisión** para lograr un equilibrio entre la complejidad del modelo y el tiempo de cómputo. Se utilizó la función `fit!` para entrenar el modelo con los datos de entrenamiento previamente preparados. El conjunto de entrenamiento incluyó las características seleccionadas (`cp`, `slope`, `thalach`) que han demostrado tener relevancia en la predicción de enfermedades cardíacas según los análisis previos. A través de este entrenamiento, el modelo fue capaz de **identificar relaciones o tendencias** en los datos, ajustando los árboles de decisión en función de las características disponibles.

En esta fase, se completó el entrenamiento del modelo y está listo para ser evaluado a través de validación cruzada y el conjunto de prueba, lo que permitirá medir su capacidad de generalización y rendimiento en datos no vistos.

4.3 Evaluación del modelo

Una vez entrenado el modelo de bosque aleatorio, el siguiente paso es evaluar su capacidad de generalización. La validación del modelo es crucial para medir su desempeño en datos no vistos y asegurar que no haya sobreajuste [9]. Para ello, se utilizan dos enfoques:

1. Validación cruzada: Permite evaluar el modelo en diferentes subconjuntos de los datos de entrenamiento, proporcionando una estimación más robusta del rendimiento del modelo.
2. Evaluación en el conjunto de prueba: Una vez validado el modelo, se utiliza el conjunto de prueba para medir su rendimiento final en datos completamente nuevos, que no se han utilizado en el entrenamiento ni en la validación.

Se emplea validación cruzada k -fold ($k=5$) para obtener una medida más confiable de la exactitud del modelo. A continuación, se presenta el código utilizado para llevar a cabo esta validación:

Algoritmo 6.

```
[116]: function cross_validation_model(model, X, y, n_folds=5)
    n = size(X, 1)
    indices = shuffle(collect(1:n))
    folds = [round(Int, (i-1)*n/n_folds+1):round(Int, i*n/n_folds) for i in 1:n_folds]
    accuracies = Float64[]

    for fold in folds
        test_idx = indices[fold]
        train_idx = setdiff(1:n, test_idx)

        # Entrenar el modelo
        DecisionTree.fit!(model, X[train_idx, :], y[train_idx])

        # Predecir con el modelo
        predictions = DecisionTree.predict(model, X[test_idx, :])

        # Calcular la exactitud
        accuracy = mean(predictions .== y[test_idx])
        push!(accuracies, accuracy)
    end

    return mean(accuracies)
end

# Usar la función
accuracy_cv = cross_validation_model(forest_model, X_train_matrix, y_train_vector)
println("Exactitud promedio de validación cruzada con el modelo existente: $accuracy_cv")

Exactitud promedio de validación cruzada con el modelo existente: 0.7560374149659864
```

Este proceso nos da una medida promedio del rendimiento del modelo, permitiendo evaluar su robustez en diferentes particiones de los datos de entrenamiento.

Una vez obtenida la validación cruzada, se procede a evaluar el modelo con el conjunto de prueba, que representa datos no vistos por el modelo. Esto nos proporcionará una medida de su capacidad de generalización.

Algoritmo 7.

```
[125]: # Realizar predicciones en el conjunto de prueba
y_pred = DecisionTree.predict(forest_model, X_test_matrix)

# Calcular la exactitud en el conjunto de prueba
test_accuracy = mean(y_pred .== y_test_vector)
println("Exactitud en el conjunto de prueba: $test_accuracy")

Exactitud en el conjunto de prueba: 0.7213114754098361
```

Se puede ver en la salida del código que el modelo presenta una exactitud de alrededor del 72%.

5 Análisis de los resultados

El modelo de clasificación basado en bosques aleatorios alcanzó una exactitud promedio del 72% en el conjunto de prueba, lo que indica un buen desempeño general en la predicción del riesgo de enfermedad cardíaca. Esta cifra es consistente con investigaciones previas que han utilizado algoritmos similares para tareas médicas de clasificación [7][10]. Si bien esta exactitud no garantiza un diagnóstico perfecto, sí representa una mejora significativa respecto a métodos tradicionales que no logran incorporar múltiples variables de forma simultánea.

Entre las variables con mayor peso en la predicción se encuentran el tipo de dolor torácico (cp), la frecuencia cardíaca máxima (thalach) y la pendiente del segmento ST (slope). Estos atributos coinciden con indicadores clínicos comúnmente utilizados por cardiólogos para evaluar riesgo cardiovascular, lo que refuerza la validez del modelo. La capacidad del algoritmo para identificar estas relaciones sin necesidad de reglas explícitas demuestra su valor como herramienta complementaria de apoyo diagnóstico.

No obstante, es importante reconocer ciertas limitaciones en el análisis. El conjunto de datos utilizado proviene del repositorio UCI, el cual, si bien ha sido ampliamente utilizado y presenta una estructura limpia y estandarizada, no está libre de sesgos de selección. El tamaño muestral relativamente reducido (303 pacientes) y la posible falta de representatividad de la población global limitan la generalización directa de los resultados. Además, la distribución de clases podría no reflejar adecuadamente la prevalencia real de la enfermedad cardíaca, lo que puede afectar el equilibrio del modelo y su precisión en casos clínicos reales.

A pesar de estas limitaciones, los resultados obtenidos son alentadores. El uso de datos secundarios permite enfocar el trabajo en la validación e implementación del modelo, lo que facilita experimentación rápida y controlada. Sin embargo, se recomienda, en investigaciones futuras, validar el modelo con datos provenientes de entornos clínicos reales, idealmente con mayor diversidad demográfica y volumen.

Finalmente, aunque se utilizó la exactitud como métrica principal, sería recomendable incorporar otras medidas como la sensibilidad, especificidad, y la curva ROC en estudios posteriores, especialmente considerando que en medicina es crucial minimizar tanto los falsos negativos como los falsos positivos.

6 Conclusiones y trabajos futuros.

El presente trabajo demuestra que el uso de técnicas de aprendizaje automático, en particular los bosques aleatorios, puede contribuir significativamente al diagnóstico de enfermedades cardíacas mediante modelos de clasificación precisos y robustos. La implementación de este enfoque permitió desarrollar una herramienta capaz de procesar múltiples variables clínicas y predecir con efectividad la probabilidad de padecer esta patología.

El modelo construido mostró un desempeño sólido en pruebas de validación, lo que refuerza su potencial como sistema de apoyo en entornos clínicos. Además, su diseño permite una interpretación razonable de las variables más influyentes en la predicción, lo cual favorece su adopción por profesionales médicos que requieren transparencia en los sistemas de inteligencia artificial.

Como líneas futuras de investigación, se sugiere ampliar el conjunto de datos con muestras más diversas y representativas de distintas poblaciones. Asimismo, sería conveniente explorar el uso de métricas adicionales como la sensibilidad y el valor predictivo positivo, para asegurar un rendimiento clínicamente relevante. Finalmente, se podrían comparar distintos algoritmos de clasificación y desarrollar prototipos de aplicación en tiempo real, con el fin de integrar este tipo de modelos en herramientas prácticas para el personal de salud.

Referencias

- [1] World Health Organization: WHO. [https://www.who.int/es/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/es/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)) (2021). Accedido el 14 de diciembre de 2024.
- [2] Breiman, L.; Friedman, J. H.; Olshen, R. A.; Stone, C. J.: Classification and regression trees. Chapman & Hall/CRC Press (1998).
- [3] Galea, A.: Beginning data science with Python and Jupyter. Packt Publishing Ltd (2018).
- [4] Govindan, G.; Hora, S.; Palagachev, K.: The data analysis workshop. Packt Publishing Ltd (2020).
- [5] Janosi, A.; Steinbrunn, W.; Pfisterer, M.; Detrano, R.: Heart Disease. UCI Machine Learning Repository. <https://archive.ics.uci.edu/dataset/45/heart+disease> . Accedido el 11 de octubre de 2024.
- [6] Nazarathy, Y.; Klok, H.: Statistics with Julia: Fundamentals for data science, machine learning and artificial intelligence. Springer Nature Switzerland AG (2021).

- [7] Kneusel, R. T.: The art of randomness. No Starch Press (2024).
- [8] Matloff, N.: The art of machine learning. No Starch Press (2024).
- [9] Zhou, Z.-H.: Machine learning. Springer Nature Singapore Pte Ltd (2021).
- [10] Geetha, T. V.; Sendhilkumar, S.: Machine learning: concepts, techniques and applications. Chapman & Hall/CRC Press (2023).
- [11] Smolyakov, V.: Machine learning algorithms in depth. Manning Publications Co. (2023).