Detección de posibles afecciones cardiovasculares empleando técnicas de machine learning

Cristian A. Bottazzi, Esteban J. Odetti, Andrés F. Pozzer
cristian.bottazzi@gmail.com, estebanode@gmail.com, andres.pozzer.ap@gmail.com
Inteligencia Computacional – Departamento de Informática – FICH UNL

***Resumen*— Las enfermedades cardiovasculares son la primera causa de muerte a nivel global con un estimado de 18 millones de muertes al año. Estas representan casi un de 31% de todas las muertes en todo el mundo[1]. Las fallas cardíacas son comúnmente causadas por enfermedades preexistentes, pudiendo ser prevenidas incorporando hábitos saludables con el fin de mitigar los factores de riesgo que predisponen a este grupo de enfermedades.**

En el presente trabajo se propone analizar el desempeño de distintos clasificadores binarios basados en aprendizaje automático aplicados a un conjunto de datos que contiene características relevantes para la detección de posibles enfermedades cardiovasculares. Dichas predicciones están realizadas sobre datos que pueden ser obtenidos de forma sencilla, rápida y en forma rutinaria en cualquier clínica u hospital.

Los pasos básicos para la preparación del sistema son: preprocesar el dataset; entrenar los distintos modelos propuestos; realizar una búsqueda de parámetros óptimos y finalmente comparar los distintos resultados.

***Palabras claves*— aprendizaje automático, enfermedad cardiovascular, detección, clasificación.**

I. INTRODUCCIÓN

Según la Organización Mundial de la Salud (OMS) las enfermedades cardiovasculares (ECV) son un conjunto de afecciones del corazón y vasos sanguíneos. Se estima que aproximadamente el 90% de las enfermedades cardiovasculares pueden ser prevenidas. La mayoría de estas enfermedades están asociadas estrechamente al modo de vida de cada individuo y en cierta medida a factores genéticos.

De acuerdo a la OMS, en 2015 murieron más de 17,7 millones de personas y más de tres cuartas partes de las defunciones se producen en los países de ingresos bajos y medios. Resulta entonces relevante la posibilidad de realizar una detección precoz y tratamiento temprano de estas afecciones, por lo que el *machine learning* será de especial ayuda en esta tarea como método de diagnóstico de soporte para el médico.

Para entrenar y testear los modelos se utilizó el dataset de Cleveland[2], el cual es uno de los conjuntos de datos más empleados para entrenar modelos relacionados con afecciones cardíacas. El mismo cuenta con 303 registros agrupados en 13 características (o *features*) con su respectiva salida esperada (enfermo/no enfermo).

En el presente trabajo se ha buscado implementar un clasificador empleando diversas técnicas y comparándolas entre sí.

Estos métodos son: Support Vector Classification (SVC), Random Forest Classifier (RFC), Multi-Layer Perceptron Classifier (MLP), K-Neighbors Classifier (KNN), Voting Ensemble Classifier (VM) soft y hard.

II. MATERIALES Y METODOLOGÍAS

En esta sección se describe el dataset utilizado y como fue normalizado. El preprocesamiento, entrenamiento y test fue realizado utilizado Python v3.9 con las bibliotecas Numpy, Sklearn, Pandas y Seaborn para la representación de los datos.

A. DATASET

El dataset está conformado por un conjunto de datos reales reunidos por la Cleveland Clinic Foundation[3] el cual fue empleado para el entrenamiento y prueba sobre la capacidad de generalización de la solución planteada con la finalidad

[1] https://www.who.int/es/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)
[2] https://archive.ics.uci.edu/ml/datasets/Heart+Disease
[3] V.A. Medical Center, Long Beach and Cleveland Clinic Foundation: Robert Detrano, M.D., Ph.D

de, posteriormente, emplear los métodos con datos nuevos. Como fue descripto en la introducción, consta de 303 entradas agrupadas en 13 características o *features*.

Cada una de estas entradas del dataset se encuentran referenciadas en la Tabla 1.

Id	Columna	Referencia
1	age	Edad en años
2	sex	Sexo. 1 = hombre. 0 = mujer.
3	cp	ChestPain - Tipo de dolor en el pecho.
4	trestbps	Presión sanguínea en reposo (mmHg)
5	chol	Colesterol sérico en mg/dl
6	fbs	Azúcar en sangre en ayuno
7	restecg	ECG en reposo (val. Entre 0 y 2)
8	thalach	Máximo ritmo cardíaco alcanzado
9	thalrest	Ritmo cardíaco en reposo
10	exang	Angina inducida por ejercicio
11	oldpeak	Depresión de la ST
12	slope	Pendiente del segmento ST
13	ca	Número de vasos mayores.
14	tal	Normal, defecto fijo, defecto reversible

Tabla 1 – Referencias para cada columna de entrada del dataset

B. Normalización del dataset

El primer lugar se lleva a cabo la homogenización de las características del conjunto de datos: La normalización implica llevar los valores de las columnas a una escala común. El objetivo principal es asociar formas similares a los mismos datos en una única forma de datos, ya que algunas columnas son binarias, otras poseen cantidades en sus respectivas unidades (mmHg, mg/dl, etc) y otras, como *thal*, poseen saltos de valores discretos.

El proceso consiste en: Para cada característica (representada como una columna en el dataset) se toma el mayor (x_{max}) y el menor (x_{min}) de los valores de entrenamiento. Y cada valor x_i (siendo i el índice de la fila) reemplazado por

$$x_i = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

III. MÉTODOS DE MACHINE LEARNING

En el presente trabajo se emplearon y compararon seis algoritmos de clasificación.

Para separar todos los pasos llevados a cabo, la Figura 1 muestra un diagrama de bloques para visualizar cada uno de los procesos que son descriptos en esta unidad.

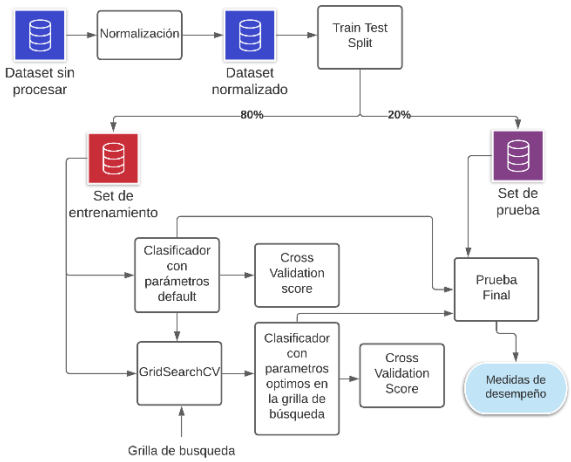

Figura 1: Diagrama de bloques para el entrenamiento y prueba de los algoritmos propuestos

A. Datos de entrada para los métodos.

Para los clasificadores seleccionados se utilizó un esquema 80-20 de los datos normalizados. Esto se realiza en la función denominada *train test split* que mezcla los datos en forma aleatoria y los divide en dos conjuntos: entrenamiento y prueba. Adicionalmente se verifica que esta división de *train* y *test* se encuentren igualmente balanceadas. En la Figura 2 se muestran la proporción de elementos balanceados en cada categoría.

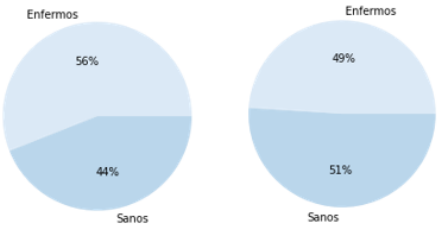

Figura 2: Izq. balance de clases en el training set y Der. balance de clases en el test set

B. Clasificadores propuestos:

I. Support vector machine (SVC): [2]

Es un método supervisado que busca el hiperplano que separe de forma óptima las clases. En sentido matemático, el algoritmo busca el máximo margen de separación entre las dos clases y construye un hiperplano en el medio de este margen máximo. La función del hiperplano está dada por:

$$y = sgn(\sum_{i=1}^{n} w_i x_i + b)$$

Donde **w** y b son los parámetros del hiperplano, y x los puntos de entrenamiento.

Para poder abordar problemas de clases no linealmente separables, la idea supone transformar el problema no lineal en lineal. Para esto existen las funciones Kernel.

II. Random Forest Classifier (RFC): [3]

Es un algoritmo de aprendizaje supervisado que consiste en conformar un gran número de árboles de decisión y someterlos a votación. La clase que resulta más votada será finalmente la ganadora.

Los árboles de decisiones son muy veloces, pero no son buenos para clasificar nuevos patrones. Es por ello que se los usa en conjunto para mejorar la *accuracy* del modelo. Resumiendo, brevemente el modelo:

1) Se crea un "*bootstrapped dataset*": Esto consiste en crear un set de datos eligiendo aleatoriamente patrones del dataset original. Este puede contener patrones repetidos.
   Con los patrones originales que no fueron seleccionados se forma otro conjunto denominado "*out of bag dataset*".
2) Se construye el árbol de decisión para ese *bootstrapped dataset*:
   Se selecciona al azar un conjunto de *features* o características que serán los nodos de decisión del árbol.
3) Retornar a paso 1 m veces. Por lo que al finalizar este bucle se tienen m arboles de decisión distintos. El fuerte componente estocástico es lo que hace fuerte al método.

Para clasificar un nuevo patrón se somete a votación a los árboles de decisión. Los *out of bag* datasets se pueden utilizar para realizar un test sobre el árbol ganador. En el presente trabajo no fue empleada esta última votación de test. En su lugar se empleó el método de validación cruzada.

III. Perceptrón multicapa (MLP): [6]

Este es el método visto en la asignatura. El modelo empleado consta de 3 capas: entrada, oculta y salida. Utilizando el método de gradiente descendiente y la retro propagación vistas en la cátedra se entrena una red neuronal con 1000 épocas y tasa de aprendizaje de 0.1.

IV. KNN Classifier:

Calcula la distancia a un nuevo punto con respecto a todos los demás puntos de entrenamiento, la distancia puede ser euclidea, etc. Luego se seleccionan los K puntos más cercanos siendo la clase a la que pertenece la mayoria de los puntos.

V. Voting Ensemble Majority: [1]

Este último método reúne las salidas de todos los modelos descriptos anteriormente para ser combinados en un sistema de votación llamado *Voting Ensemble Majority*. Aquí las salidas de cada clasificador son sometidas a votación para seleccionar la etiqueta de clasificación definitiva del elemento.

Existen dos tipos:

1) **Voting Majority Hard (VMH):**
   Todos los clasificadores tienen el mismo peso y la salida es la clase con mayor cantidad de votos.
2) **Voting Majority Soft (VMS):**
   Clasifica según las probabilidades que cada método le otorga a la clase. Luego aplicando una media aritmética entre todas las probabilidades de las clases se determina cual es la salida.

C. Búsqueda de parámetros óptimos:

Luego de entrenar cada clasificador con parámetros por defecto (no hubo ningún ajuste sobre los parámetros de entrada), se empleó la función *GridSearchCV* con la finalidad de buscar los mejores parámetros dentro de un conjunto prefijado. La comparación utiliza una validación cruzada de k-fold con $k = 5$.

IV. RESULTADOS

Como primera medida, para evitar que un método tenga mejor inicialización que otro, se fijó el valor de la semilla *random state* = 0. De cambiarse el valor de inicialización, se debe realizar una nueva búsqueda de los parámetros.

Como método de verificación de los parámetros hallados se realizó la función de validación cruzada "*cross val score*" sobre los datos de entrenamiento. Se empleó la estrategia de k-fold con $k = 5$.

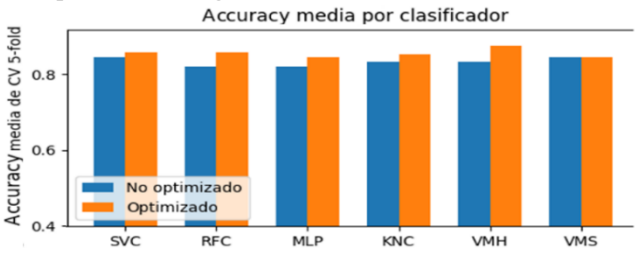

Fig. 3: accuracy media del k-fold optimizados vs no optimizados

La Figura 3 muestra el resultado de la accuracy media obtenido para cada uno de los métodos empleados. Se observa que, en general, hay un ligero aumento de la misma. Principalmente para los métodos de Voting Majority en su versión *Hard* y para el RFC, que rondan una mejora del ~6,5%. El Voting Majority Soft, por otro lado, no obtuvo mejoras.

Método	Accuracy	Sensibilidad	Especificidad
SVM	**0.77**	**0.79**	**0.75**
RFC	**0.84**	**0.96**	**0.70**
MLP	**0.80**	**0.85**	**0.76**
KNC	**0.77**	**0.84**	**0.72**
VMH	**0.77**	**0.79**	**0.75**
VMS	**0.80**	**0.85**	**0.76**

Tabla 2: Medidas de desempeño en test set.

La Tabla 2 muestra que la mayoría de los clasificadores catalogan de manera correcta un individuo sano con una media de 0,75. Además, clasifican de manera correcta a los individuos enfermos en un 0,82 de media, exceptuando el RFC que obtuvo un valor de 0,96 de sensibilidad; y la precisión de los métodos ronda el 0,8 de media, una vez más favoreciendo el RFC. La matriz de confusión para este método se muestra en la Figura 4.

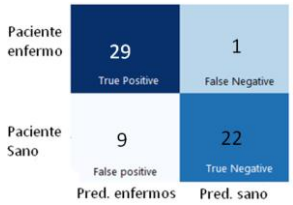

Figura 4: Matriz de confusión para el RFC

En las siguientes imágenes se presentan las curvas ROC para cada uno de los métodos sin optimizar (Figura 5) y optimizados (Figura 6), exceptuando el VMH que no está representado debido a que este método carece del uso de probabilidades para catalogar las clases.

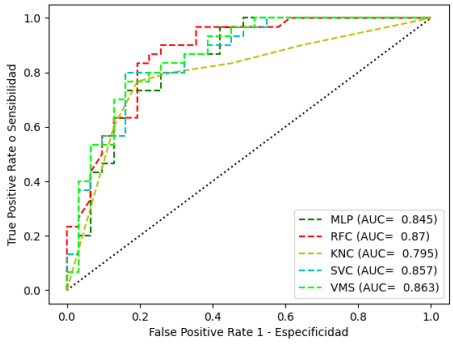

Figura 5: Curva ROC con parámetros no optimizados

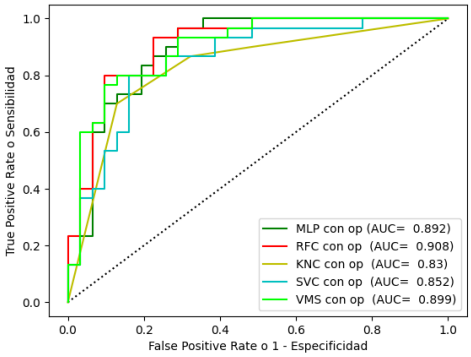

Figura 6: Curva ROC con parámetros optimizados

Se construye variando el umbral de probabilidad lo cual nos da 21 valores distribuidos entre 0 y 1, dando así distintos valores de especificidad y sensibilidad. Al inicio obtenemos baja sensibilidad, pero alta especificidad y al final obtenemos alta sensibilidad, pero baja especificidad.

V. CONCLUSIONES

Como se observa en la figura 5 y 6 las áreas bajo las curvas de los respectivos métodos aumentan, esto confirma una mejora al optimizar los parámetros ya que al aumentar el área aumenta la especificidad y la sensibilidad, con su correspondiente baja en falsos positivos y los falsos negativos.

A partir de la tabla 2 podemos concluir que el test es muy bueno para el VMS, pero no obtuvimos mejores resultados que con el RFC.

Es posible concluir entonces que a partir del número de casos y por los valores devueltos por cada clasificador, el método que ha mostrado ser más efectivo es el Random Forest Classifier (RFC) a pesar de que su potencial se maximiza para grandes datasets[4]. En este caso a pesar de que solo se lo ha entrenado con 303 patrones se obtuvieron resultados satisfactorios. Adicionalmente, se minimizó el sesgo durante el entramiento asegurando que las clases se encontraran apropiadamente balanceadas como se ha observado en la Figura 2.

Por último, en el VMH, la accuracy media en el *cross val score* fue la más alta, a pesar de que con los datos de prueba no fue posible obtener mejores resultados que el mejor clasificador (RFC). Lo importante desde nuestro punto de vista es que se clasifique de manera correcta a los enfermos ya que si a un sano lo clasifica de manera incorrecta el mismo se puede realizar un chequeo medico para confirmar, en cambio con los pacientes enfermos corre mayores riesgos.

VI. REFERENCIAS

[1] R. Atallah and A. Al-Mousa, "Heart Disease Detection Using Machine Learning Majority Voting Ensemble Method," 2019 2nd International Conference on new Trends in Computing Sciences (ICTCS), 2019, pp. 1-6, doi: 10.1109/ICTCS.2019.8923053.

[2] Miloš Marjanović, Miloš Kovačević, Branislav Bajat, Vít Voženílek, Landslide susceptibility assessment using SVM machine learning algorithm, Engineering Geology, Volume 123, Issue 3, 2011, ISSN 0013-7952, https://doi.org/10.1016/j.enggeo.2011.09.006.

[3] Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. R news, 2(3), 18-22.

[4] Khennou, Fadoua & Fahim, Charif & Chaoui, Habiba & Nour El Houda, Chaoui. (2019). A Machine Learning Approach: Using Predictive Analytics to Identify and Analyze High Risks Patients with Heart Disease. International Journal of Machine Learning and Computing. 9. 762-767. 10.18178/ijmlc.2019.9.6.870.

[5] Al-Milli, Nabeel. (2013). Backpropogation neural network for prediction of heart disease.

[6] Leandro Ezequiel Di Persia, Matias Fernando Gerard, Diego Humberto Milone, Jose Tomas Molas Gimenez, Hugo Leonardo Rufiner, Georgina Silvia Stegmayer, Leandro Daniel Vignolo, *Apuntes de catedra*.

[4] An overtraining-resistant stochastic modeling method for pattern recognition