

ORIGINAL

Comparison of kernel functions in the prediction of cardiovascular disease in Artificial Neural Networks (ANN) and Support Vector Machines (SVM)

Comparativo de funciones kernel en la predicción de enfermedades cardiovasculares en Redes Neuronales Artificiales (ANN) y Máquinas de Soporte Vectorial (SVM)

Michael Rafael Rodríguez Rodríguez¹ ✉, Claudia Alejandra Delgado Calpa¹ ✉, Héctor Andrés Mora Paz¹

¹Universidad CESMAG, Facultad de Ingeniería, Ingeniería de Sistemas. Pasto, Colombia.

Citar como: Rodríguez Rodríguez MR, Delgado Calpa CA, Mora Paz HA. Comparison of kernel functions in the prediction of cardiovascular disease in Artificial Neural Networks (ANN) and Support Vector Machines (SVM). EthAlca. 2025; 4:172. <https://doi.org/10.56294/ai2025172>

Enviado: 16-06-2024

Revisado: 08-11-2024

Aceptado: 03-07-2025

Publicado: 04-07-2025

Editor: PhD. Rubén González Vallejo 

Autor para la correspondencia: Michael Rafael Rodríguez Rodríguez ✉

ABSTRACT

Cardiovascular diseases are currently the leading cause of death worldwide. There are challenges, such as untimely healthcare, lack of access to technologies and timely diagnoses. Therefore, this project focuses on the use of innovative tools, giving way to the need to use artificial intelligence in the field of Machine Learning to improve the prediction of cardiovascular diseases. The research focused on determining the most effective kernel function in Artificial Neural Network (ANN) and Support Vector Machine (SVM) algorithms, making a fair comparison and evaluating the accuracy and prediction time of each proposed kernel function. Based on the results, these new optimal kernel functions are integrated into the scikit-learn library, achieving validation in the appropriate configuration for predicting the risk of CVD. This innovative approach reduces detection time, minimising the chances of future complications from preventable diseases, and provides timely diagnosis and risk factors with early warnings that can be extremely useful for healthcare personnel.

Keywords: Kernel Functions; Prediction; Cardiovascular Diseases; Artificial Neural Networks; Vector Support Machines; Machine Learning; Artificial Intelligence.

RESUMEN

En la actualidad las enfermedades cardiovasculares, constituyen la principal causa de muerte a nivel mundial. Existen desafíos, como la inoportunidad en la atención en salud, falta de acceso a las tecnologías y diagnósticos oportunos, por ende, este proyecto se enfoca en el uso de herramientas innovadoras, dándole paso a la necesidad de utilizar inteligencia artificial en el ámbito de Machine Learning, para mejorar la predicción de las enfermedades cardiovasculares, es así que la investigación se centró en determinar la función kernel más eficaz en algoritmos de Redes Neuronales Artificiales (ANN) y Máquinas de Soporte Vectorial (SVM), haciendo un comparativo ecuánime, evaluando la exactitud y el tiempo de predicción de cada función kernel propuesta. Con base en los resultados, se integran esas nuevas funciones kernel optimas a la biblioteca scikit-learn, logrando una validación en la configuración apropiada para la predicción del riesgo de padecer alguna ECV, este enfoque innovador permite reducir el tiempo de detección, minimizando así las posibilidades de complicaciones futuras de enfermedades que se pueden prevenir, y aportar de manera oportuna en el diagnóstico y factores de riesgo con alertas tempranas que pueden ser de gran utilidad para el personal de salud.

Palabras clave: Funciones Kernel; Predicción; Enfermedades Cardiovasculares; Redes Neuronales Artificiales; Máquinas de Soporte Vectorial; Machine Learning; Inteligencia Artificial.

INTRODUCCIÓN

Las enfermedades cardiovasculares son desórdenes del corazón y de los vasos sanguíneos, y se han convertido en la principal causa de muerte en todo el mundo.^(1,2,3,4,5,6,7,8,9,10,11) En Colombia, durante 2021, se registraron un total de 51 988 muertes a raíz de condiciones relacionadas con ECV, un 12 % más que en el 2020, con un aumento significativo en la incidencia de casos en mujeres, de acuerdo con cifras del Departamento Nacional de Estadísticas (DANE) 4. A pesar de que el país ha avanzado en los esfuerzos para reducir la carga de estos escenarios, aún se presentan acciones muy fragmentadas dentro del sistema de salud y los sacrificios individuales de cada actor no son suficientes.^(2,6,12,13,14,15,16,17,18,19,20) Existen diferentes elementos que pueden estar relacionados con la ineficiencia en la prevención oportuna de estas patologías, como puede ser la inadecuada atención en los centros de salud, difícil acceso a tecnologías, inoportunidad en los diagnósticos, escasa información sobre factores modificables que logran afectar directamente su salud y llegar a complicaciones severas a futuro.^(20,21,22,23,24,25,26,27,28,29,30)

Por tales motivos, los profesionales sanitarios se ven obligados a cambiar la forma de abordar los padecimientos de los pacientes desde los principios de la promoción de la salud y la prevención de la enfermedad como una nueva política de salud pública.^(2,3,31,32,33,34,35,36,37,38,39,40)

En otras palabras, como se menciona por Arias⁽⁴⁾, aun si se logra asegurar el acceso a un determinado servicio de salud, especialmente en los sectores más vulnerables, el beneficio es nulo si no se tiene la capacidad de salvar vidas y mejorarlas con los recursos que se destinan para estas enfermedades, en las que una monitorización y análisis fuera del entorno clínico continúa siendo fundamental para descubrir oportunamente las complicaciones asociadas en gran parte de la población. Una rápida actuación en cualquier lugar resulta vital, es esencial reducir el tiempo de detección y atención, de forma que se minimicen las posibilidades de sufrir daños cardiovasculares, caer en estado de coma, o, en el peor de los escenarios, la muerte.^(41,42,43,44,45,46,47,48,49,50,51)

Existen tecnologías que prometen revolucionar el ámbito de la prevención de las enfermedades cardiovasculares. Así mismo, gracias a sus grandes capacidades, son la herramienta perfecta para mejorar su predicción.

En consecuencia, se han puesto en marcha diferentes investigaciones cuyos resultados determinan que el Machine Learning podría ayudar muchísimo en la determinación de sufrir episodios o enfermedades cardiovasculares en individuos sanos.^(5,52,53,54,55,56,57,58,59,60)

Lo que se pretende con la siguiente investigación es determinar de las funciones algebraicas y trascendentes, la función kernel que ofrece el mejor compromiso en la predicción de enfermedades cardiovasculares en los algoritmos de SVM y ANN, aportando a los investigadores con las funciones kernel, a partir de un estudio comparativo ecuánime, donde se mida la exactitud y el tiempo de predicción de cada función; posteriormente se acoplan estas nuevas funciones kernel a la biblioteca Scikit-learn, se valida la configuración tomando la más apropiada para la prevención de ECV teniendo en cuenta la relación error tiempo, y logrando así construir la aplicación móvil para alertas tempranas implementando el modelo predictivo.^(61,62,63,64,65,66,67,68,69,70,71,72)

De un conjunto de funciones trascendentes ¿Cuál función kernel tiene el mejor compromiso entre la exactitud y el tiempo de predicción, para el apoyo al diagnóstico de enfermedades cardiovasculares?

Objetivo General

Determinar de las funciones algebraicas y trascendentes, la función kernel que ofrece el mejor compromiso en la predicción de enfermedades cardiovasculares en los algoritmos de SVM y ANN a partir de un estudio comparativo ecuánime, donde se mida la exactitud y el tiempo de predicción de cada función kernel.

MÉTODO

Paradigma

Esta investigación se enmarca dentro del paradigma positivista, un paradigma que es racional, objetivo y se basa en comprobar hechos y particularidades propias del conocimiento científico, característico de la esencia disciplinar ingenieril, a su vez se caracteriza por el alto interés, por la verificación del conocimiento a través de predicciones. Algunos lo llaman el “paradigma prediccionista”, ya que lo importante es plantearse una serie de hipótesis como predecir que algo va a suceder y luego verificarlo o comprobarlo. En las ciencias exactas y naturales es en donde tiene mayor aplicación.⁽³⁸⁾ El presente proyecto permite estudiar el desempeño de los algoritmos SVM y ANN a través del uso de las funciones kernel, realizando varias configuraciones de hiperparámetros a través de una sintonización sistemática, aplicando tales métodos con datos. Además, que la presencia de las variables en los datasets influye de manera significativa para una posterior predicción y posterior análisis.

Enfoque

El enfoque a usar en esta investigación es empírico-analítico, es un método de observación utilizado para

profundizar en el estudio de los fenómenos, pudiendo establecer leyes generales a partir de la conexión que existe entre la causa y el efecto en un contexto determinado, además es uno de los modelos para describir el método científico, que se basa en la experimentación y la lógica empírica.⁽³⁹⁾ Aborda la realidad de los hechos que son observables, cuantificables y medibles. Es un método que contrasta sus hipótesis de una forma rigurosa a través de la demostración científica que determina si dicha hipótesis es verdadera o falsa.⁽³⁸⁾ El actual proyecto permite observar y analizar las causas y efectos de la problemática estudiada, plantear procesos de experimentación y pruebas, que servirán de base para proponer una solución. De esta manera se obtendrá modelos apropiados en la predicción de enfermedades cardiovasculares.

Método

El método será cuasiexperimental, porque su objetivo es poner a prueba una hipótesis causal manipulando una variable independiente para posterior a ello analizar las consecuencias que la manipulación tiene sobre una o más variables dependientes.

Esto se verá reflejado en el apoyo al diagnóstico clínico en la predicción de enfermedades cardiovasculares, fomentando a su vez de hiperparámetros para la obtención de predicciones en los algoritmos SVM, ANN frente al conjunto de datos de factores de enfermedades cardiovasculares.

La metodología que se utilizara en el proyecto es CRISP-DM, que son las siglas de Cross-Industry Standard Process for Data Mining, es un método probado para orientar trabajos de minería de datos. Como metodología, incluye descripciones de las fases normales de un proyecto, las tareas necesarias en cada fase y una explicación de las relaciones entre las tareas.⁽⁴⁰⁾



Fuente: <https://www.iic.uam.es/innovacion/metodologia-crisp-dm-ciencia-de-datos/>

Figura 1. Esquema del ciclo CRISP-DM estándar

La metodología CRISP-DM establece un proyecto de minería de datos como una secuencia de fases:⁽⁴¹⁾

1. Comprensión del negocio: el objetivo de esta fase es alinear los objetivos del proyecto de data mining con los objetivos del negocio.
2. Comprensión de los datos: dos puntos clave en esta fase: conocer los datos, estructura y distribución, y la calidad de los mismos.
3. Preparación de los datos: el objetivo final de esta fase es obtener los datos finales sobre los que aplicarán los modelos: El objetivo último de esta fase es construir un modelo que nos permita alcanzar los objetivos del proyecto.
4. Modelado: el objetivo último de esta fase es construir un modelo que nos permita alcanzar los objetivos del proyecto.
5. Evaluación: en esta fase nos centramos en evaluar el grado de acercamiento del modelo a los objetivos del proyecto.
6. Despliegue: el objetivo último de esta fase es realizar el despliegue de los resultados obtenidos de forma que sea propagado a los usuarios finales, así como el mantenimiento del mismo una vez el despliegue haya finalizado.

De momento en la actualidad los proyectos que tienen que ver con desarrollos de modelos inteligentes se concentran en dos metodologías específicas una es KDD y otra es CRISP-DM donde esta última es una metodología más dinámica, que permite la interacción del ciclo del modelo y también del software, logrando un desarrollo interactivo, permitiendo así que el modelo se pueda escalar, tiene ese dinamismo de retroalimentación que tiene de algunas fases que componen la metodología.

Tipo de investigación

El tipo de investigación que se desarrolla a lo largo del proyecto es determinado por el tipo de investigación aplicada. Se persigue la creación de conocimiento aplicándolo directamente a los problemas sociales. Se basa en los resultados tecnológicos de la investigación básica, que se ocupa de la relación entre concepto y producto.

Diseño de investigación

El diseño de investigación es de ensayos debido a que se realizarán distintos tipos de pruebas, como el análisis en cuanto a rendimiento, la confianza por el entrenamiento de algoritmos otorgando una predicción, el proyecto se basa en dos herramientas que tenga en cuenta las etapas de proceso de máquinas de soporte vectorial y redes neuronales artificiales, de esta manera se plantea las siguientes fases.

- Conexión a los datos: principalmente es necesario tener un repositorio que nos provee de datos de diferentes autores, permitiendo así una estimación de los datos que se obtienen sean suficientes y relevantes para construcción del modelo.
- Definición de un criterio de evaluación: se trata de una medida de error, comúnmente el uso del error cuadrático medio que permita su uso en medidas como la precisión y exhaustividad.
- Preparación de los datos: datos completos, combinación de datos de varias fuentes, formato adecuado a los datos, tener presente las características relevantes, y posteriormente la construcción del modelo.
- Análisis de errores: usar modelos de SVM Y ANN diferentes, revisar los datos si se requieren en su totalidad o solo se toma algunos.
- Fase de entrenamiento: al seleccionar el tipo de neurona artificial a utilizar en una red neuronal y a su vez determinada la topología, inicia el proceso de entrenamiento de la red para que pueda ser utilizada. a partir de un conjunto de pesos sinápticos aleatorios, el proceso de aprendizaje busca un conjunto de pesos que permitan a la red realizar correctamente una función específica.
- Fase de operación: una vez finalizada la fase de aprendizaje, el modelo puede que se ajuste demasiado a las particularidades presentes en los patrones de entrenamiento, perdiendo su habilidad de generalizar su aprendizaje a casos nuevos.⁽⁴²⁾
- Análisis de resultados y descubrimiento de conocimiento: Se analizarán los resultados de los métodos mediante visualizaciones, de tal forma que pueda determinar qué métodos tuvieron un mejor rendimiento en los diferentes conjuntos de datos, esto permite que los investigadores prueben sus métodos en SVM, ANN y permita así una predicción segura de las enfermedades cardiovasculares y que esta herramienta permita ser usada como ayudas diagnósticas.

Población

Esta investigación tiene como fin el estudio de la parte epistemológica y matemática del objeto de estudio, que en este caso son las funciones kernel aplicadas a SVM y ANN, por lo tanto, no se requiere población ni muestras, como se menciona en la investigación de distinción máxima Evaluación De Métodos De Reducción De Dimensión Para La Preservación Topológica De Los Datos Mediante Métricas Rnx desarrollada por (Correa Lozano.C.D., Lozano Thomé.J.A., Urrea Burgos.D.F.), ya que la naturaleza del proyecto exige un tipo de investigación de Data Driven Approach, en el cual las bases de datos son los principales en el desarrollo del proyecto. Los Datasets que se trabajaran serán probados en diferentes conjuntos de datos, en este proyecto no se utilizara datos focales, sino que se hará uso de diferentes bases de datos ya preestablecidos por otros investigadores.

Técnicas de recolección de la información

Este proyecto no requiere la recolección de información, ya que existen bases de datos con características estándar que se usan por diferentes investigadores para probarlos con diferentes métodos de predicción de riesgo cardiovascular. Pero se tendrá en cuenta las bases de datos utilizadas relacionadas a continuación:

Herramientas de repositorio de datasets Kaggle

Un instituto nacional encargado de la evaluación de la educación, con el objetivo de facilitar el uso y consulta de las bases de datos de los resultados de las pruebas que aplica, tiene a disposición de la comunidad académica el FTP digital para investigadores. Esta herramienta es un repositorio para que grupos de investigación

y estudiantes de niveles avanzados, interesados en participar en convocatorias de investigación, tengan la información de primera mano para desarrollar sus propuestas. A partir del cruce de bases disponibles, se logra recopilar información que permite clasificar al estudiante en diferentes niveles socioeconómicos. Es de interés en este proyecto predecir el nivel socioeconómico del estudiante a partir de la información de la base de datos.

Para lo anterior, use todas las técnicas de predicción vistas (algunas por ver) en clase:

- Knn - Vecinos más cercanos.
- Árboles de clasificación.
- Redes Neuronales.
- Random Forest.
- Boosting, Gradient Boosting.
- LDA, QDA, FDA.⁽⁴³⁾

Herramientas de repositorio de datasets UCI

El repositorio de aprendizaje automático de UCI es una colección de bases de datos, teorías de dominio y generadores de datos que utiliza la comunidad de aprendizaje automático para el análisis empírico de los algoritmos de aprendizaje automático. El archivo fue creado como un archivo ftp en 1987 por David Aha y sus compañeros estudiantes de posgrado en UC Irvine. Desde entonces, ha sido ampliamente utilizado por estudiantes, educadores e investigadores de todo el mundo como fuente principal de conjuntos de datos de aprendizaje automático. Como indicación del impacto del archivo, ha sido citado más de 1000 veces, lo que lo convierte en uno de los 100 “artículos” más citados en toda la informática. La versión actual del sitio web fue diseñada en 2007 por Arthur Asunción y David Newman, y este proyecto es en colaboración con Rexa.info en la Universidad de Massachusetts Amherst. Se agradece el apoyo financiero de la National Science Foundation.⁽⁴⁴⁾

Validez de la técnica

Las variables que van a tener el conjunto de datos de los repositorios van a ser preprocesados con algoritmos de limpieza, transformadores de escalamiento y algoritmos de reducción de dimensión. Para almacenar los resultados de cada experimentación que se va a realizar, en especial para las funciones kernel, se almacenara en un dataframe cuyos datos van a ser aterrizados en un archivo CSV.

Confiabilidad de la técnica

Este proyecto como se mencionó en secciones anteriores adopta técnicas de aprendizaje automático para generar datos que puedan reducir significativamente el potencial margen de error. Esto se logrará ajustando la configuración del algoritmo de manera apropiada. Además, se utilizarán los resultados obtenidos de estas pruebas para validar o refutar la hipótesis planteada.

RESULTADOS

A continuación, se presenta en este capítulo los resultados de la investigación que corresponden al desarrollo de cada uno de los objetivos planteados.

Definición de funciones Kernel recomendadas por expertos

Resultados para la definición de funciones kernel con recomendación de expertos

Para ejecutar el resultado del primer objetivo, que es definir funciones Kernel con base en la recomendación de expertos, se procedió a la búsqueda de dos documentos importantes teniendo una temática diferente entre sí, y a la vez una estrecha relación con el uso de funciones Kernel su comportamiento con variedad de datos y cuáles al final son las más adecuadas para acoplarse al modelo creado. Se comienza así con una revisión de la literatura científica relacionada con el uso de funciones Kernel, citando a estos expertos, se demuestra que existe una base teórica para explorar funciones Kernel alternativas.

Uno de los artículos de los cuales se toma en cuenta para el desarrollo de esta investigación es el artículo de Mora.H., en su proyecto titulado “Comparación de funciones Kernel para la predicción de la oferta energética fotovoltaica” publicado en el año 2020, España⁽⁴⁵⁾, y el otro experto Belanche.L., en su proyecto titulado “Kernel Functions for Categorical Variables with Application to Problems in the Life Sciences” publicado en el año 2013, Barcelona, Spain⁽⁴⁶⁾, se puede evidenciar que hacen uso de funciones Kernel alternativas, generalmente en las implementaciones de las bibliotecas que hay en el aprendizaje automático y que a su vez se implementa 4 algoritmos de funciones Kernel Lineal, Polinómica, RBF, Sigmoide, tomando este último Belanche.L que con su proyecto presenta que el enfoque propuesto puede superar notablemente a los Kernel Estándar, por lo que puede ser utilizado como una buena alternativa a otras funciones comunes del Kernel (al menos para la clasificación SVM) para obtener una mayor exactitud, con funciones alternativas se puede adquirir mejores resultados que la RBF, ya que se destaca que el Kernel Categórico aprovecha información que no es utilizada por los otros Kernels, como k0 o el Kernel RBF. Esto sugiere que el Kernel Categórico es capaz de capturar relaciones

y patrones en los datos que los otros Kernels no pueden capturar.

Es importante destacar también que en el documento se menciona que en casos en los que los resultados de los Kernels Estándar son buenos, el Kernel Categórico K1, mejora ligeramente en comparación con K0, y ambos superan al Kernel RBF. Esto indica que el enfoque propuesto es beneficioso incluso cuando los resultados ya son aceptables. Se menciona que el Kernel RBF muestra un bajo rendimiento en algunos casos, posiblemente debido a su susceptibilidad al sobre ajuste en datos de muestra pequeña en representaciones de alta dimensión.

Y el primer artículo mencionado de Mora, H., utiliza un par de funciones Kernel; la Gaussiana y Polinómica, de esta investigación se destaca que los Kernel alternativos solo han mejorado los resultados para SVM y han empeorado para ANN. Se tiene la hipótesis de que los Kernels funcionan mejor para SVM que en otros algoritmos de clasificación y predicción lineal, porque este algoritmo utiliza el criterio del margen máximo, abriendo la discusión para realizar nuevos experimentos teniendo en cuenta más algoritmos de aprendizaje automático que usan funciones Kernel, realizando dicha evaluación con distintas bases de datos para clasificación y predicción. Los resultados del análisis reflejan que SVM con los normalizadores Standard y Min-Max y los Kernel Triangular y Rational Quadratic son los más indicados para realizar regresión sobre los datos Landsat y MODIS. A su vez se tiene en cuenta las funciones Kernel que se proponen tanto en el de Belanche como en el de Mora son las que están definidas en la siguiente figura.

Kernel	
Gaussiano (RBF)	Triangular (tr)
$e^{-\sum_{i=1}^d \gamma (x_i - x'_i)^\beta}$ <p>$\gamma > 0, \beta \in (0, 2]$</p>	$\begin{cases} \ x - x'\ \leq a \rightarrow 1 - \frac{\ x - x'\ }{a} \\ \ x - x'\ > a \rightarrow 0 \end{cases}$ <p>$a > 0$</p>
Radial Basic (Rb)	Rational Quadratic (Rq)
$\frac{d}{(\sum_{i=1}^d e^{\gamma (x_i - x'_i)^2})^m}$ <p>$\gamma > 0, m \in \mathbb{N}$</p>	$1 - \frac{\ x - x'\ ^2}{\ x - x'\ ^2 + a}$ <p>$a > 0$</p>
Canberra (Can)	Truncated (Tru)
$1 - \frac{1}{d} \sum_i \gamma \frac{ x_i - x'_i }{ x_i + x'_i }$ <p>$\gamma \in (0, 1]$</p>	$\frac{1}{d} \sum_i \max(0, \frac{ x_i - x'_i }{\gamma})$ <p>$\gamma > 0$</p>

Fuente: comparativo de funciones kernel sobre predicción de oferta de fuentes alternativas de energía
Figura 2. Formulación matemática de las funciones kernel utilizadas por expertos

Y de la cual adicionalmente basada en la definición de belanche:

$$k(x, x') = \langle \emptyset(x), \emptyset(x') \rangle$$

De la cual se propone como exploración los siguientes kernels:

- Un kernel Logarítmico Radial Basic: Es una función de similitud que se utiliza para medir la distancia entre puntos de datos en un espacio de características. El kernel RBF puede mapear los datos a un espacio de características de mayor dimensionalidad donde los datos pueden ser más fácilmente separables mediante un hiperplano. Esto permite a los algoritmos como las SVM realizar clasificación y regresión en conjuntos de datos que no son linealmente separables en su espacio original. Dado que el kernel RBF puede capturar patrones no lineales, es útil en situaciones donde los datos tienen relaciones complejas y no lineales.⁽⁴⁷⁾

$$b \text{ kln rbf} = \ln \sum \gamma \langle x, x' \rangle$$

$$b \text{ kln rb} = \sum \ln \gamma \langle x, x' \rangle$$

Todo este análisis anteriormente dicho nos llevó a definir los Kernel inspirados en la concepción de funciones de activación alternativas en redes neuronales y son:

Kernel	
Polynomial (pk)	Triangular (tk)
$K(x, x') = (x, x' + c)^d$	$\begin{cases} x - x' \leq a \rightarrow 1 - \frac{ x - x' }{a} \\ x - x' > a \rightarrow 0 \end{cases}$ $a > 0$
Hyperbolic (hk)	Rational Quadratic (rqk)
$K(x, y) = \tanh(\alpha x, y + c)$	$1 - \frac{ x - x' ^2}{ x - x' ^2 + a}$ $a > 0$
Radial Basic (rbk)	Truncated (trk)
$\left(\sum_{i=1}^d e^{-\gamma(x_i - x'_i)^2} \right)^m$ $\gamma > 0, m \in \mathbb{N}$	$\frac{1}{d} \sum_i \max\left(0, \frac{ x_i - x'_i }{\gamma}\right)$ $\gamma > 0$
Canberra (ck)	Laplacian (lpl)
$1 - \frac{1}{d} \sum_i \gamma \frac{ x_i - x'_i }{ x_i + x'_i }$ $\gamma \in (0, 1]$	$k(x, y) = e^{\left(\frac{- x - y }{\gamma}\right)}$
Sigmoid (sigm)	Cosine (cos)
$k(x, y) = \tanh(\gamma x, y)$	$k(x, y) = \frac{ x - y }{ x \cdot y }$

Figura 3. Formulación matemática de las funciones kernel utilizadas en el proyecto

Se representa, a su vez, en la figura 3 la función matemática del Kernel y en la la figura 4 su respectiva codificación en el lenguaje Python.

Codificación de funciones Kernel

Por lo tanto, las funciones de activación de las redes neuronales están diseñadas para activar y representar características particulares en los datos. En el proyecto de predicción del riesgo cardiovascular, se diseñó funciones kernel alternativas que capturen patrones específicos en los datos clínicos o de salud relacionados con factores de riesgo cardiovascular, como presión arterial, peso, talla, índice de masa corporal entre otros, una vez diseñados, estos Kernels alternativos requerirían optimización y ajuste en función de conjuntos de datos específicos de esta manera se podría evaluar si estos Kernels mejoran la capacidad de predicción y generalización en comparación con los Kernels tradicionales en modelos de aprendizaje automático.

Para llevar a cabo la codificación de esas funciones kernel se tuvo en cuenta la figura 5 que hace relación al algoritmo de comparación utilizando conjunto de datos.

Se realiza una configuración manual primero se define: la carpeta de entrada, aquí están los datos a inspeccionar, la carpeta de salida es donde se colocaron los resultados de la comparación y el tipo de problema a procesar para este caso es de clasificación. Luego, se obtiene un vector de funciones del kernel y normalizadores, se procede a la lectura de los datos ubicados en el directorio de entrada, seguido de la ejecución de las siguientes actividades:

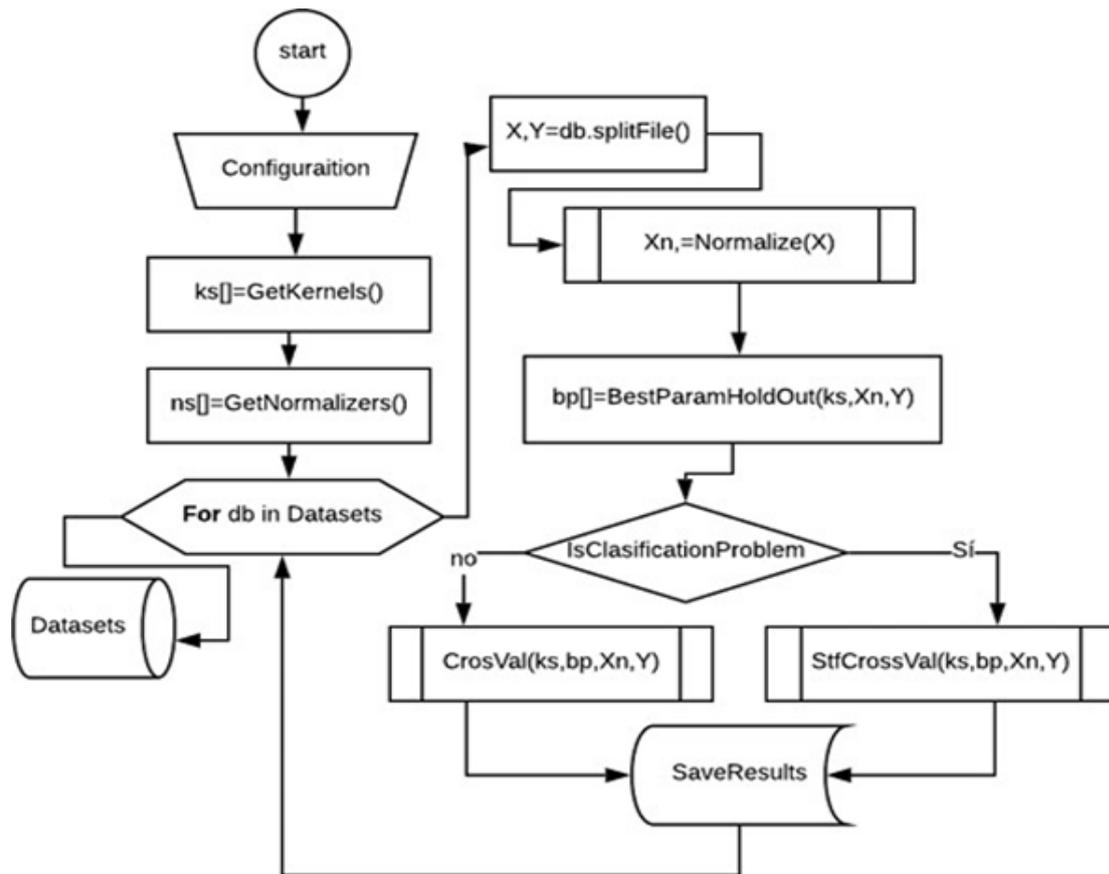
- Se realiza una partición de los datos respectivamente.
- Se realiza la normalización de los datos y se identifican los parámetros donde el Kernel respectivo obtuvo un puntaje superior o un menor error.
- Se realiza una validación con estos parámetros, como el problema es de clasificación, se realiza una validación cruzada dividida; cuando finalizan las pruebas, los resultados se registran en la carpeta de resultados.

Kernel	Kernel
Polynomial (pk)	Triangular (tk)
<pre>#POLYNOMIAL KERNEL m E N, a>0 def polynomial(x,y, degree=3, gamma=0.01,coef0=0.0): m=degree a=gamma return (a*np.dot(x,y)+1)**m</pre>	<pre>#Triangle a>0 def triangle(x,y, degree=3, gamma=0.01,coef0=0.0): a=gamma norm=la.norm(np.subtract(x, y)) if norm<=a: return 1-norm/a return 0</pre>
Hyperbolic (hk)	Rational Quadratic (rqk)
<pre>#Hyperbolic tangent kernel a0>0, b<0 def hyperbolic(x,y, degree=3, gamma=0.01,coef0=0.0): b=coef0 a=gamma return np.tanh(a*np.dot(x,y)+b)</pre>	<pre>#Rational quadratic a>0 def rquadratic(x,y, degree=3, gamma=0.01,coef0=0.1): a=coef0 norm=la.norm(np.subtract(x, y)) return 1-(norm**2)/(norm**2+a)</pre>
Radial Basic (rbk)	Truncated (trk)
<pre>def radial_basic(x, y, degree=3, gamma=0.01,coef0=0.0): m=degree sm=0 sm=np.sum(np.exp(-gamma*((x-y)**2))) return sm**m</pre>	<pre>#Truncated gamma>0 def truncated(x,y, degree=3, gamma=0.01,coef0=0.0): sm=0 d=x.shape[0] val=1-np.abs(x-y)/gamma sm=np.sum(val[val>0]) return sm/d</pre>
Canberra (ck)	Laplacian (lpl)
<pre>#CANNBERRA def c_canberra_kernel(gamma): def ck(X,Y): return canberra(X,Y,gamma=gamma) return ck</pre>	<pre>#LAPLACIAN def c_laplacian_kernel(gamma): def lpl(X,Y): return laplacian(X,Y,gamma=gamma) return lpl</pre>
Sigmoid (sigm)	Cosine (cos)
<pre># sigmoid def sigmoid(x,y, degree=3, gamma=0.01,coef0=0.0): #print('x',x,'y',y) sm=np.tanh(gamma*x@y+coef0)</pre>	<pre>def cosine(x,y, degree=3, gamma=0.01,coef0=0.0): #print('x',x,'y',y) sm=(x@y)/(np.linalg.norm(x)*np.linalg.norm(y)) if np.isnan(sm) or np.isinf(sm): sm=0 return sm</pre>

Figura 4. Codificación de funciones kernel en python

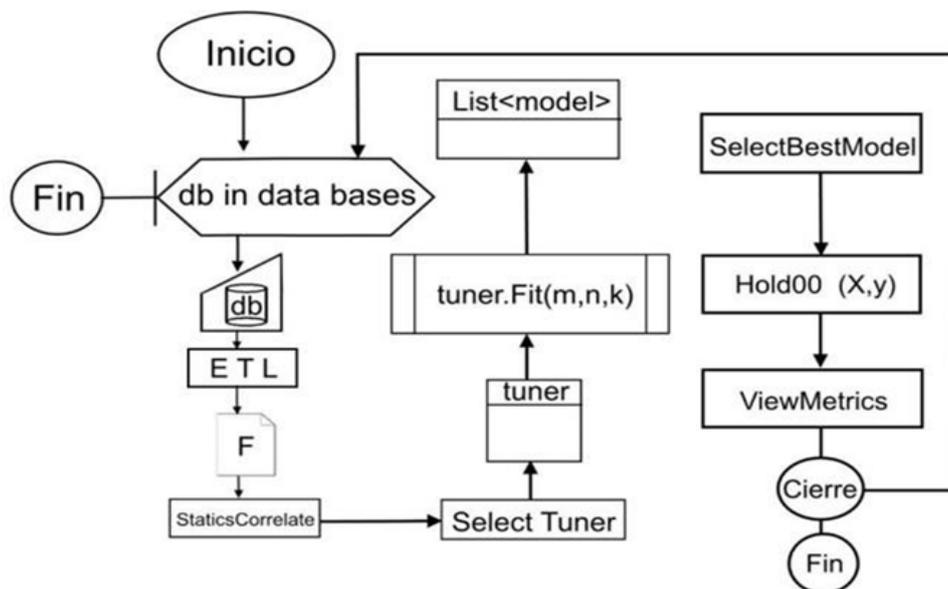
Este método inicial proporciona una estimación aproximada de qué función del Kernel tiene el mejor rendimiento, pero para obtener una aproximación más precisa, se requiere una sintonización de hiperparámetros utilizando técnicas avanzadas, por lo que se establece el siguiente algoritmo.

En la figura 6 se muestra que primero se ejecuta un ciclo para recorrer la base de datos, extrayendo, transformando y cargando datos, los cuales finalmente se sintetizan en un archivo en la que se encuentran las estadísticas y aspectos de correlación, después de elegir un sintonizador predeterminado, configurar un motor de búsqueda evolutivo (algoritmo genético), este sintonizador selecciona la mejor configuración de parámetros, los mejores son seleccionados y reentrenados.



Fuente: <https://docplayer.es/192405564-Comparativo-de-kernels-sobre-prediccion-de-oferta-de-fuentes-alternativas-de-energia.html>

Figura 5. Algoritmo implementado para la comparación de kernel



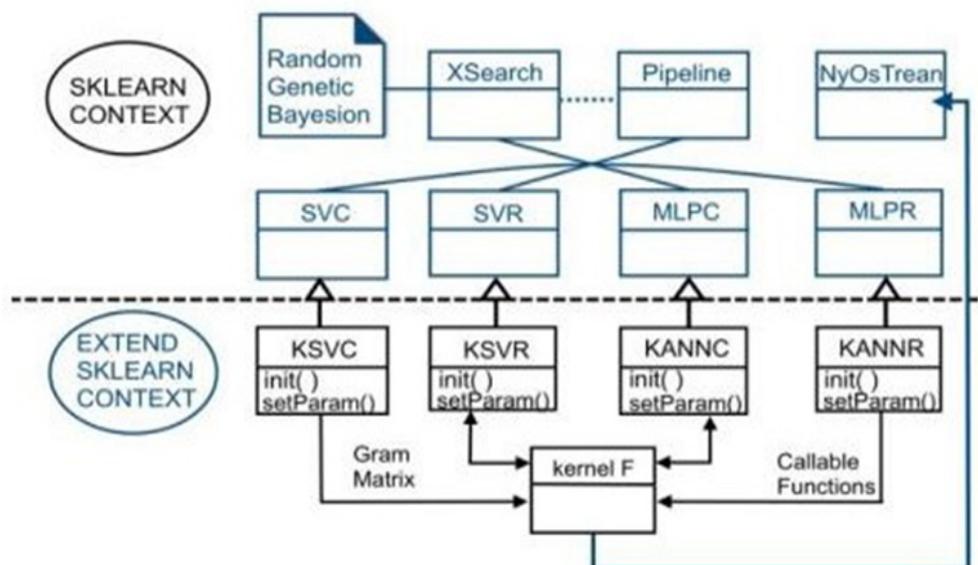
Fuente: <https://docplayer.es/192405564-Comparativo-de-kernels-sobre-prediccion-de-oferta-de-fuentes-alternativas-de-energia.html>

Figura 6. Algoritmo exhaustivo para búsqueda de modelos

Funciones Kernel acopladas a SVM y ANN

Acoplamiento con biblioteca Scikit-learn

Para llevar a cabo y cumplir con este segundo objetivo de acoplar nuevas funciones Kernel alternativas en los algoritmos SVM, ANN a la biblioteca Scikit-learn. Se tomó como base lo siguiente:



Fuente: <https://docplayer.es/192405564-Comparativo-de-kernels-sobre-prediccion-de-oferta-de-fuentes-alternativas-de-energia.html>

Figura 7. Modelo de acoplamiento funciones Kernel en Scikit-learn

El truco Kernel se introduce en SVM para clasificación en la clase SVC y se nutre de más funciones kernel con la clase KSVC mediante la matriz de Gram como se muestra en las ecuaciones (1) y (2):

$$\max: L(a) = \sum_1^n a_i - \frac{1}{2} \sum_{i,j=1}^n a_i a_j y_i y_j k(x_i, x_j) \quad (1)$$

$$\text{s. a: } \sum_1^n a_i y_i = 0, 0 \leq a_i \leq C, i = 1, 2, \dots, n \quad (2)$$

Un proceso similar se sigue para el algoritmo de regresión provista con la clase KSVR cuya expresión formal se muestra en las ecuaciones (3) y (4):

$$\max: L(a) = \sum_1^n (a_i^- - a_i^+) y_i - \epsilon \sum_1^n (a_i^- + a_i^+) - \frac{1}{2} \sum_{i,j=1}^n (a_i^- - a_i^+) (a_j^- - a_j^+) k(x_i, x_j) \quad (3)$$

$$\text{s. a: } \sum_1^n (a_i^- - a_i^+) = 0, 0 \leq (a_i^-, a_i^+) \leq C, i = 1, 2, \dots, n \quad (4)$$

Para ANN se utilizó la aproximación de Nystroem (SI et al., 2017) mediante una transformación de base sobre el vector de características. Formalmente:

$$Z_L = W_L \cdot a_{L-1} + b_L \quad (5)$$

$$N_t = \text{NyosTream}(kf) \quad (6)$$

$$X = N_t * X \quad (7)$$

$$a_0 = X \quad (8)$$

$$a_L = f(Z_L) \quad (9)$$

$$\hat{y} = \text{predict}(X_{batch} * N_t^T) \quad (10)$$

Donde la ecuación (5) representa la suma ponderada, mientras que (6) la transformación de Nystroem preparada con una función Kernel. La ecuación (7) es la generación de la transformación, conjuntamente (8) es la activación inicial o capa de entrada y (9) la función de activación que se ha definido lineal por defecto, buscando que las funciones kernel sean las que cumplan el oficio de la transformación del espacio de características, en este caso un espacio de Hilbert de dimensión infinita. Cabe resaltar que es necesario sobrescribir el método predict del clasificador o regresor en KANNC y KANNR para realizar la transformación sobre cada batch del descenso del gradiente estocástico SGD (10).

En la siguiente figura se muestra cómo se acopla la clase KSVC a la biblioteca Scikit-learn la cual sigue un procedimiento análogo para KSVR, KANNC, KANNR.

```
class KSVC(SVC):

    def __init__(self, C=1.0, kernel='rbf', degree=2, gamma='auto',
                 coef0=0.0, shrinking=True, probability=False,
                 tol=1e-3, cache_size=200, class_weight=None,
                 verbose=False, max_iter=-1, decision_function_shape='ovr',
                 random_state=None, a=2):

        super().__init__(
            kernel=kernel, degree=degree, gamma=gamma,
            coef0=coef0, tol=tol, C=C, shrinking=shrinking,
            probability=probability, cache_size=cache_size,
            class_weight=class_weight, verbose=verbose, max_iter=max_iter,
            decision_function_shape=decision_function_shape,
            random_state=random_state)

        self.a=a

    def fit(self, X, y, sample_weight=None):

        if(self.kernel=="linear" or self.kernel=="poly" or self.kernel=="rbf"):
            super().fit(X,y)
            return self
        else:
            if(self.kernel=="mrbf"):
                self.kernel=mrbf_kernel(degree=self.degree, gamma=self.gamma)
                super().fit(X,y)
                return self
            if(self.kernel=="hyperbolic"):
                self.kernel=hyperbolic_kernel(self.gamma,self.coef0)
                super().fit(X,y)
                return self
            if(self.kernel=="triangle"):
                self.kernel=triangle_kernel(self.gamma)
                super().fit(X,y)
                return self
            if(self.kernel=="radial_basic"):
                self.kernel=radial_basic_kernel(self.degree, self.gamma)
                super().fit(X,y)
                return self
```

Figura 8. Código fuente de acoplamiento a biblioteca Scikit-learn

```

if(self.kernel=="rquadratic"):
    self.kernel=rquadratic_kernel(self.degree, self.gamma, self.coef0)
    super().fit(X,y)
    return self
if(self.kernel=="can"):
    self.kernel=canberra_kernel(self.gamma)
    super().fit(X,y)
    return self
if(self.kernel=="tru"):
    self.kernel=truncated_kernel(self.gamma)
    super().fit(X,y)
    return self
if(self.kernel=="chisq"):
    self.kernel=additive_chi2_kernel
    super().fit(X,y)
    return self
if(self.kernel=="chi2"):
    self.kernel=chi2_kernel
    super().fit(X,y)
    return self
if(self.kernel=="laplacian"):
    self.kernel=laplacian_kernel
    super().fit(X,y)
    return self
if(self.kernel=="sigmoid"):
    self.kernel=sigmoid_kernel
    super().fit(X,y)
    return self
if(self.kernel=="cosine"):
    self.kernel=cosine_similarity
    super().fit(X,y)
    return self

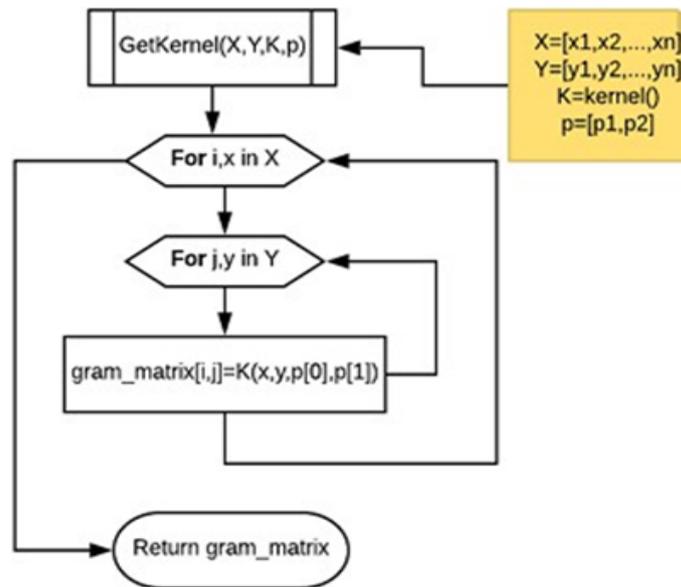
def set_params(self, **parameters):
    for parameter, value in parameters.items():
        setattr(self, parameter, value)
    return self

```

Figura 9. Código fuente de acoplamiento a biblioteca Scikit-learn

Obtención de funciones Kernel (procedimiento GetKernels)

Para obtener este procedimiento se construyó la matriz de Gram correspondiente a la función Kernel K (figura 10) tomando como Kernel la formulación de la figura 15 enviando los parámetros correspondientes a cada Kernel.



Fuente: <https://docplayer.es/192405564-Comparativo-de-kernels-sobre-prediccion-de-oferta-de-fuentes-alternativas-de-energia.html>

Figura 10. Algoritmo para la obtención de funciones kernel

Para los recursos realizados en la figura 11 bastaría con enviar como parámetro al algoritmo de aprendizaje (SVM, ANN).

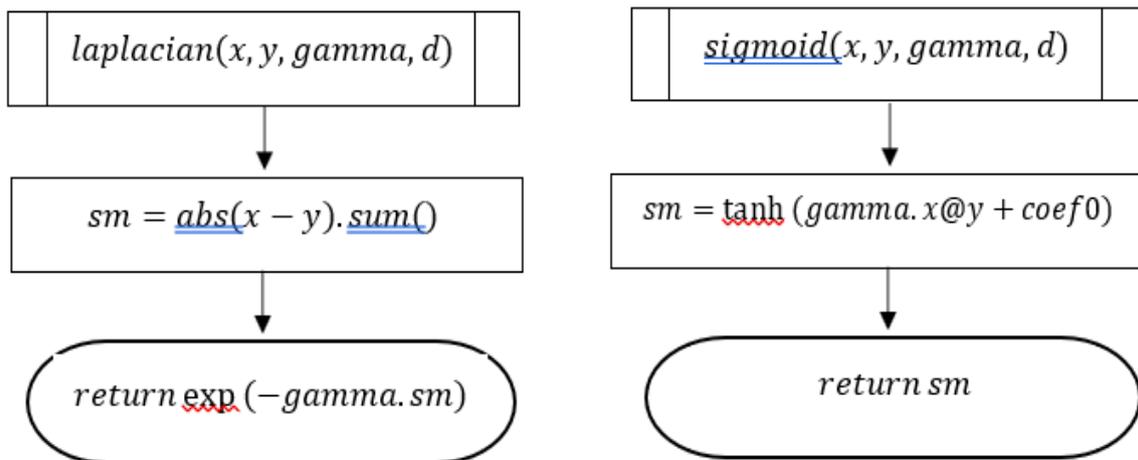


Figura 11. Ejemplo de implementación de un kernel L laplacian(izq.), S sigmoid (der)

Sin embargo, en la mayoría de los casos, estos algoritmos están diseñados para recibir una función en su inicializador (constructor) y enviar los datos de entrenamiento X, Y (funciones de retorno).

Validando la configuración apropiada de los Kernel

Con el fin de alcanzar el tercer objetivo, que consiste en validar la configuración más apropiada para la prevención de enfermedades cardiovasculares mediante la comparación de la relación entre el error y el tiempo se registraron los resultados de cada experimento en un dataframe, para su posterior análisis.

Descripción de los datos

La base de datos que se usó para el desarrollo del proyecto fue un dataset que consta de 70 000 registros de datos de pacientes, contiene 11 características más objetivo, y a su vez consta de 3 tipos de entradas:

- Objetivo: información fáctica.
- Examen: resultados de examen médico.
- Subjetivo: información dada por el paciente.

En la figura 12 se puede observar las características de los datos.

Edad	Característica de objetivo	años
Altura	Característica de objetivo	Cm
Peso	Característica de objetivo	Kg
Género	Característica de objetivo	Código categórico 1: mujer 2: hombre
Presión arterial sistólica	Función del examen	Ap_hi
Presión arterial diastólica	Función del examen	Ap_lo
Colesterol	Función del examen	1: normal 2: por encima de lo normal 3: muy por encima de lo normal
Glucosa	Función del examen	1: normal
		2: por encima de lo normal 3: muy por encima de lo normal
Fumar	Característica subjetiva	Si el paciente fuma 1: Si 2: No
Consumo de alcohol	Característica subjetiva	Alco Binario 1: Si 2: No
Actividad física	Característica subjetiva	Activo Binario 1: Si 2: No
Presencia o ausencia de enfermedad cardiovascular	Variable objetivo	Cardio Binario Variable Objetivo 1: Presencia de ECV 2: Ausencia de ECV

Figura 12. Descripción de los datos

Obtención del mejor parámetro de entrenamiento para SVM y ANN

En esta fase, se emplea un método de búsqueda exponencial para explorar los diferentes valores de los parámetros asociados a un Kernel específico dentro de un rango definido. Posteriormente, se lleva a cabo una división de los datos en conjuntos de entrenamiento y prueba, reservando una muestra del 14,29 %. Una vez completada esta etapa, se calcula la puntuación del modelo utilizando los datos de prueba, identificando así los mejores parámetros y su correspondiente puntuación óptima.

Resultados inspección de funciones Kernel en clasificación para SVM

En la figura 13 se presentan los resultados de la evaluación de las funciones en problemas de clasificación para determinar la adecuación de las funciones Kernel. Este análisis busca validar la afirmación de Mora.H., y Belanche.L., la cual sugiere que las funciones consideradas en este estudio pueden conducir a resultados mejorados.

En la figura 13 se detallan los resultados de la validación mediante la división en conjuntos de entrenamiento y prueba realizada con el algoritmo SVM para clasificación, donde se evaluaron diversas funciones Kernel. Este

proceso se llevó a cabo utilizando el conjunto de datos previamente mencionado y aplicando técnicas de normalización y escalación.

SCALER	KERNEL	MEAN_TEST_SCORE	BEST_PARAM
SScaler	rquadratic	0.710	γ : auto coef0: 545.5594 C: 46.415
SScaler	rbf	0.698	γ : 0.0006 C: 2154.4346
SScaler	tru	0.718	γ : 0.0001 C: 1.0
SScaler	can	0.717	γ : 0.0006 C: 2154.4346
SScaler	radial_basic	0.708	γ : 0.0006 C: 2154.4346
SScaler	triangle	0.712	γ : 545.5594 C: 46.4158
SScaler	hyperbolic	0.703	γ : 0.0006 C: 2154.4346
SScaler	laplacian	0.718	γ : 0.0001 C: 1.0
SScaler	sigmoid	0.601	γ : 0.0001 C: 1.0
SScaler	cosine	0.708	C: 46.4158
MMScaler	rquadratic	0.706	γ : auto coef0: 11.2883 C: 2154.4346
MMScaler	rbf	0.696	γ : 0.0006 C: 2154.4346
MMScaler	tru	0.718	γ : 0.0001 C: 1.0
MMScaler	can	0.717	γ : 0.0006 C: 2154.4346
MMScaler	radial_basic	0.703	γ : 0.0006 C: 2154.4346

MMScaler	triangle	0.704	γ : 4.2813 C: 12.9154
MMScaler	hyperbolic	0.692	γ : 0.0006 C: 2154.4346
MMScaler	laplacian	0.716	γ : 0.0001 C: 1.0
MMScaler	sigmoid	0.658	γ : 4.2813 C: 12.9154
MMScaler	cosine	0.715	C: 2154.4346
NMScaler	rquadratic	0.716	γ : auto coef0: 0.2335 C: 7742.6368
NMScaler	rbf	0.717	γ : 11.2883 C: 2154.4346
NMScaler	tru	0.702	γ : 0.0048 C: 3.5938
NMScaler	can	0.701	γ : 4.2813 C: 12.9154
NMScaler	radial_basic	0.722	γ : 11.2883 C: 2154.4346
NMScaler	triangle	0.71	γ : 11.2883 C: 2154.4346
NMScaler	hyperbolic	0.54	γ : 0.2335 C: 7742.6368
NMScaler	laplacian	0.703	γ : 0.0006 C: 2154.4346
NMScaler	sigmoid	0.536	γ : 0.2335 C: 7742.6368
NMScaler	cosine	0.718	C: 100000.0

Figura 13. Resultados generales para svm

En la figura 13 se han resaltado los Kernels que mejor score obtuvieron y a su vez el comparador de escalación y el Kernel correspondiente.

Este comparador de escalación toma el mejor score de la figura 12 y lo reevalúa por validación cruzada, para este caso la función Kernel Truncated. En general para este conjunto se produjeron muy buenos resultados a excepción del Kernel Hyperbolic, por lo que se visualizara las salidas del normalizador Normalizer Scaler, Standard Scaler ya que producen mejores resultados en cada Kernel.

Por consiguiente, se puede notar que el valor del parámetro alcanza un score máximo, con este parámetro se realiza una validación cruzada con diferentes métricas. En la figura 14 se puede observar los parámetros que se usan para la próxima etapa del algoritmo.

KERNEL	SCALER	BEST_PARAM	VALOR	SCORE
rbf	SScaler	C γ	2154.4346 0.0006	0.698
Can	SScaler	C γ	2154.4346 0.0006	0.717
Tru	SScaler	C γ	1.0 0.0001	0.718
hyperbolic	SScaler	C γ	2154.4346 0.0006	0.703
triangle	SScaler	C γ	46.4158 545.5594	0.712
Radial basic	SScaler	C γ	2154.4346 0.0006	0.708
Rquadratic	SScaler	C γ Coef0	46.4158 auto 545.5594	0.710
Laplacian	SScaler	C γ	1.0 0.0001	0.718
Sigmoid	SScaler	C γ	11.0 0.0001	0.601
Cosine	SScaler	C	46.4158	0.708
rquadratic	NMScaler	C γ Coef0	7742.6368 auto 0.2335	0.716
rbf	NMScaler	C γ	2154.4346 11.2883	0.717
tru	NMScaler	C	3.5938	0.702

		γ	0.0048	
can	NMScaler	c	12.9154	0.701
		γ	4.2813	
radial_basic	NMScaler	c	2154.4346	0.722
		γ	11.2883	
triangle	NMScaler	c	2154.4346	0.71
		γ	11.2883	
hyperbolic	NMScaler	c	7742.6368	0.54
		γ	0.2335	
laplacian	NMScaler	c	2154.4346	0.703
		γ	0.0006	
sigmoid	NMScaler	c	7742.6368	0.536
		γ	0.2335	
Cosine	SScaler	c	100000.0	0.708

Figura 14. Mejores parámetros configurados para las funciones kernel en svm

Clasificación de los resultados

Los resultados finales de los Kernel previa configuración de los mejores parámetros descritos anteriormente se sintetiza en la figura 15, donde se muestra el Kernel y la métrica de exactitud (Accuracy). Esta métrica indica la proporción de predicciones correctas realizadas por el modelo en comparación con el total de predicciones, de esta manera se valida lo expuesto por los expertos Mora.H., y Belanche.L., en la que se afirma que las funciones a considerar en el presente estudio pueden obtener mejores resultados para las Máquinas de Soporte Vectorial, culminando así la primera fase experimental.

Kernel	Exactitud (Accuracy)
rbf	0.58
can	0.725
tru	0.71
hyperbolic	0.445
triangle	0.61
radial basic	0.685
Rquadratic	0.605
Laplacian	0.575
Sigmoid	0.445
Cosine	0.705

Figura 15. Resultados finales para svm

Se observa en la figura 15 que la mayor exactitud la obtuvo el Kernel Canberra para este conjunto de datos. Con una exactitud promedio del 72 %.

Resultados inspección de funciones Kernel en clasificación para ANN

SCALER	KERNEL	MEAN_TEST_SCORE	BEST_PARAM
SScaler	rquadratic	0.684	γ : auto coef0: 4.2813 C: 12.9154
SScaler	rbf	0.703	γ : 0.2335 C: 7742.6368
SScaler	tru	0.702	γ : 0.2335 C: 7742.6368
SScaler	can	0.712	γ : 0.2335 C: 7742.6368

SScaler	radial_basic	0.716	γ : 4.2813 C: 12.9154
SScaler	triangle	0.704	γ : 4.2813 C: 12.9154
SScaler	hyperbolic	0.688	γ : 4.2813 C: 12.9154
SScaler	laplacian	0.703	γ : 0.2335 C: 7742.6368
SScaler	sigmoid	0.703	γ : 545.5594 C: 46.4158
SScaler	cosine	0.704	C: 1.0
MMScaler	rquadratic	0.661	γ : auto coef0: 0.0018 C: 3.5938
MMScaler	rbf	0.658	γ : 11.2883 C: 2154.4346
MMScaler	tru	0.712	γ : 0.2335 C: 7742.6368
MMScaler	can	0.73	γ : 206.9138 C: 3.5938
MMScaler	radial_basic	0.711	γ : 4.2813 C: 12.9154
MMScaler	triangle	0.636	γ : 4.2813 C: 12.9154
MMScaler	hyperbolic	0.622	γ : 4.2813 C: 12.9154
MMScaler	laplacian	0.688	γ : 0.2335 C: 7742.6368
MMScaler	sigmoid	0.618	γ : 0.0048 C: 3.5938
MMScaler	cosine	0.618	C: 100000.0
NMScaler	rquadratic	0.528	γ : auto coef0: 545.5594 C: 46.4158
NMScaler	rbf	0.7	γ : 3792.6901 C: 27825.5940

NMScaler	tru	0.681	γ : 0.0048 C: 3.5938
NMScaler	can	0.687	γ : 4.2813 C: 12.9154
NMScaler	radial_basic	0.643	γ : 3792.6901 C: 27825.5940
<i>NMScaler</i>	<i>triangle</i>	<i>0.711</i>	γ : 0.0048 C: 3.5938
NMScaler	hyperbolic	0.522	γ : 0.0006 C: 2154.4346
NMScaler	laplacian	0.703	γ : 545.5594 C: 46.4158
NMScaler	sigmoid	0.522	γ : 0.0006 C: 2154.4346
NMScaler	cosine	0.528	C: 27825.5940

Figura 16. Resultados generales para ann

En la figura 16, se presentan los resultados de la evaluación de las funciones en problemas de clasificación para ANN para determinar la adecuación de las funciones Kernel. Este análisis busca validar la afirmación por los expertos Mora.H., y Belanche.L., donde se afirma que las funciones a considerar empeoran los resultados para ANN.

Se detallan los resultados de la validación mediante la división en conjuntos de entrenamiento y prueba realizada con el algoritmo ANN para clasificación, donde se evaluaron diversas funciones Kernel. Este proceso se llevó a cabo utilizando el conjunto de datos previamente mencionado y aplicando técnicas de normalización y escalación.

En la figura 17 se han resaltado los Kernel que mejor score obtuvieron y a su vez el comparador de escalación y el Kernel correspondiente.

Este comparador toma el mejor score de la figura 16 y lo reevalúa por validación cruzada para este caso la función Kernel Radial Basic. En general, para este conjunto no se produjeron muy buenos resultados a excepción del Kernel mencionado, por lo que se visualizara solo las salidas del Standard Scaler ya que produce mejores resultados en cada Kernel.

Por consiguiente, se puede notar que el valor del parámetro alcanza un score máximo, con este parámetro se realiza una validación cruzada con diferentes métricas. En la figura 14 se puede observar los parámetros que se usan para la entrada para la próxima etapa del algoritmo.

KERNEL	BEST_PARAM	VALOR	SCORE
rquadratic	c	12.9154	0.684
	γ	auto	
	Coef0	4.2813	
rbf	c	7742.6368	0.703
	γ	0.2335	

tru	c	7742.6368	0.702
	γ	0.2335	
can	c	7742.6368	0.712
	γ	0.2335	
<i>radial_basic</i>	c	12.9154	0.716
	γ	4.2813	
triangle	c	12.9154	0.704
	γ	4.2813	
hyperbolic	c	12.9154	0.688
	γ	4.2813	
laplacian	c	7742.6368	0.703
	γ	0.2335	
sigmoid	c	46.4158	0.703
	γ	545.5594	
cosine	c	1.0	0.704

Figura 17. Mejores parámetros configurados para las funciones kernel en ann

Clasificación de los resultados

Los resultados finales de los Kernel previa configuración de los mejores parámetros descritos anteriormente se sintetiza en la figura 18, donde se muestra el Kernel y la métrica de exactitud (Accuracy). Esta métrica indica la proporción de predicciones correctas realizadas por el modelo en comparación con el total de predicciones, de esta manera se valida lo expuesto por los expertos Mora.H., y Belanche.L., en la que se afirma que las funciones a considerar en el presente estudio pueden obtener mejores resultados para las Máquinas de Soporte Vectorial, pero no para las redes neuronales artificiales culminando así la primera fase experimental.

Kernel	Exactitud (Accuracy)
rbf	0.445
can	0.695
tru	0.67
hyperbolic	0.445
triangle	0.445
radial basic	0.645
Rquadratic	0.445
Laplacian	0.445
Sigmoid	0.445
Cosine	0.445

Figura 18. Resultados finales para ann

Se observa en la figura 18 que la mayor exactitud la obtuvo el Kernel Canberra para este conjunto de datos. Con una exactitud promedio del 69 %.

Una vez obtenidos los resultados de experimentación tanto para SVM y ANN se creó el modelo con el Kernel que mejor Score tuvo, en este caso y para estos datos es más favorable utilizar algoritmos de clasificación basados en SVM dado su rendimiento.

Construir una aplicación web para alertas tempranas de enfermedades cardiovasculares, a través de un modelo predictivo.

Construcción de la herramienta de software

Dado que puede haber usuarios que no tengan habilidades de programación y, por lo tanto, no puedan utilizar el modelo para evaluar los Kernel, se ha desarrollado una herramienta que proporciona un entorno interactivo y fácil de usar para realizar la evaluación de los nuevos Kernels inspirados en la metodología y mediante, esta se pueda hacer una evaluación del riesgo cardiovascular y obtener una alerta temprana.

La metodología que se implementó en todo el desarrollo de la investigación y de los objetivos es la metodología CRISP-DM, no fue necesario una metodología adicional para desarrollar el software porque este perfectamente hace parte de la etapa del despliegue, puesto que el aplicativo lo único que va a realizar es consumir el modelo y mostrar la interfaz sencilla al usuario final, para ello se desarrolló varias actividades presentadas a continuación.

Diagrama arquitectónico

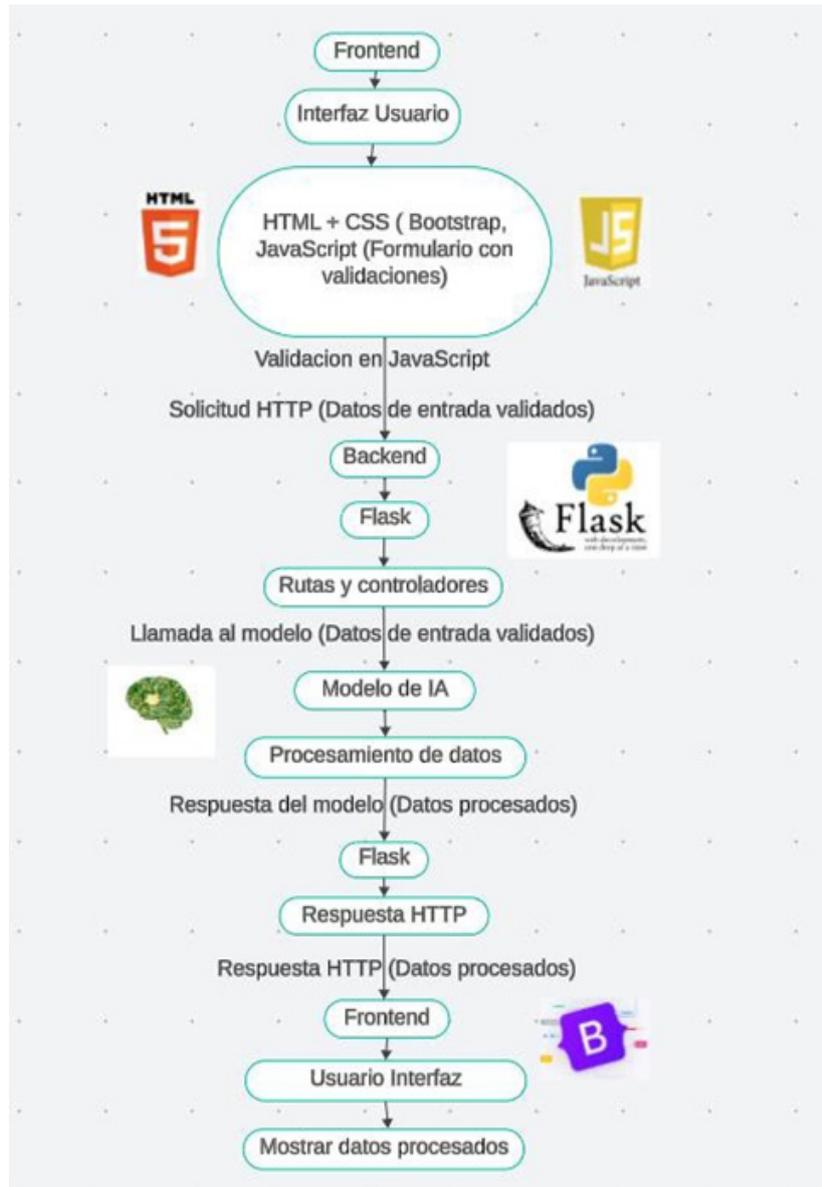


Figura 19. Diagrama arquitectónico

Arquitectura

Interacción del usuario: el usuario interactúa con la interfaz, construida con HTML y Bootstrap, introduciendo datos en un formulario.

Validación en JavaScript: antes de enviar el formulario, JavaScript valida los datos introducidos. **Envío del formulario:** Una vez validados los datos, el formulario se envía al backend usando el atributo action del formulario.

Procesamiento en Flask

- Flask recibe la solicitud y extrae los datos.
- Flask envía estos datos al modelo de IA para su procesamiento.

Modelo de IA: el modelo de IA procesa los datos y devuelve los resultados a Flask.

Respuesta al frontend

- Flask toma los resultados del modelo de IA y crea una respuesta HTTP.
- Esta respuesta se envía de vuelta al frontend.

Mostrar resultados: el navegador recibe la plantilla HTML con los resultados procesados y los muestra al usuario.

Creación del modelo

En esta etapa, se construye el modelo predictivo utilizando técnicas de aprendizaje supervisado, específicamente utilizando un algoritmo de Máquina de Soporte Vectorial (SVC) con el Kernel Truncated. Se seleccionan las técnicas de modelado apropiadas para este caso basándose en las características de los datos y los requisitos del problema. Como se mencionó anteriormente, se ha optado por utilizar un modelo de Máquina de Soporte Vectorial (SVC) debido a su capacidad para manejar conjuntos de datos lineales como no lineales, y su eficacia en problemas de clasificación. Además, al habilitar la probabilidad (probability=True), el modelo proporciona estimaciones de probabilidad para las predicciones, lo cual es útil en la evaluación de riesgos como lo es en nuestro caso.

Abstracción del escenario

Dentro del marco de este proyecto centrado en el riesgo cardiovascular, el escenario de enfoque en la salud se establece en la identificación, prevención y gestión de las enfermedades cardiovasculares. Este entorno de investigación aplica a toda la población como individuos con factores de riesgo, personas con antecedentes cardiovasculares, y aquellos que podrían beneficiarse de intervenciones preventivas y que no padezcan de alguna enfermedad cardiovascular. Además, se considera la calidad de la atención médica, la disponibilidad de tecnologías de monitoreo y las barreras que enfrentan las personas en la detección temprana y prevención de eventos cardiovasculares. El objetivo es mejorar la salud cardiovascular a través de la aplicación de modelos de inteligencia artificial y el análisis de datos para la predicción temprana y la implementación de estrategias preventivas.

Visualización de instrucciones

En esta sección se establecen descripciones útiles que permitan una mayor comprensión del aplicativo web y sus funciones.

Análisis: requerimientos funcionales para la visualización de instrucciones.

<i>RF-01</i>	Visualizar instrucciones
<i>Versión:</i>	1.0 (01-04-2024)
<i>Autores:</i>	Michael Rafael Rodríguez Rodríguez Claudia Alejandra Delgado Calpa
<i>Descripción:</i>	La aplicación web debe mostrar al usuario las instrucciones de uso del formulario y su entorno virtual.
<i>Importancia:</i>	Esencial
<i>Urgencia:</i>	Inmediata
<i>Estabilidad:</i>	Alta
<i>Estado:</i>	Implementado
<i>Comentarios:</i>	Ninguno

Figura 20. Requerimiento funcional rf-01

Desarrollo

La herramienta cuenta con instrucciones claras y concisas para guiar al usuario a través del proceso de ingreso de datos y visualización de resultados. Estas instrucciones son fáciles de entender por qué genera ejemplos de medición y está disponible en un formato accesible para todos los usuarios.

CRK

Instrucciones para el uso del aplicativo

1. Sobre CRK CardioRiskMeter:

Esta aplicación contiene un modelo entrenado con el mejor kernel que permite la identificación del riesgo cardiovascular identificando a su vez aquellos factores de riesgo predominantes logrando brindar alertas tempranas.

2. Instrucciones para el uso de la aplicación:

En el primer módulo se encuentra un formulario con varios ítems de los cuales están disponibles para recolectar la información que ud proporcione, es importante recordar que debe contar con resultados previos de glucosa en ayuno y

Ruta de Atención Clínica:

La Ruta de Atención Clínica es fundamental para la implementación efectiva de la Iniciativa CRK(CardioRiskMeter), mejorando la gestión clínica y autoconocimiento. Además, capacita al usuario y a su familia al educarlos sobre el manejo completo de los factores de riesgo cardiovascular. Esta aplicación abarca la ruta de atención clínica para el manejo integral del riesgo cardiovascular.

3. Acerca de CRK(CardioRiskMeter):

CRK es una iniciativa que surge de un proyecto de grado dirigida a fortalecer los sistemas de atención primaria de salud. Su enfoque se centra en mejorar la prevención y el control de las enfermedades cardiovasculares y sus factores de riesgo. Esta iniciativa tiene como meta fomentar un proceso de mejora continua en la calidad de atención. Esto se logra mediante un mejor control de la hipertensión y la implementación de un enfoque integral en el manejo del riesgo cardiovascular.

4. Uso:

En ninguna circunstancia esta aplicación pretende sustituir la consulta con un profesional de la salud ni su criterio clínico. Su objetivo es proporcionar un apoyo al diagnóstico médico, una herramienta para evaluar rápidamente el riesgo cardiovascular y facilitar la comunicación con los usuarios sobre las posibles modificaciones. Además, busca asistir a las personas preocupadas por su salud, ayudándoles a determinar cuándo es necesario buscar atención médica si su riesgo no es bajo. Las recomendaciones en hábitos de vida saludables están descritas a nivel de información y no deben ser utilizadas como guía sin la supervisión de un personal de salud, ya que esto podría resultar peligroso.

Evaluación del riesgo cardiovascular:

La evaluación del riesgo cardiovascular comienza por la identificación de esos datos proporcionados. Estas indagaciones sobre: Ingresar el género, la edad, tabaquismo, actividad física, presencia o ausencia de alguna enfermedad cardiovascular, peso, altura, los valores de colesterol total y presión arterial sistólica y diastólica, glucosa de ayuno. El dato de riesgo cardiovascular será un porcentaje visible mediante un diagrama de barras que indicará la presencia del riesgo de padecer alguna enfermedad cardiovascular.

Modificación del riesgo corrigiendo factores:

Una vez obtenida la estimación de riesgo, se desplegarán unas alertas tempranas que nos permiten identificar el factor de riesgo que se ve alterado, para que se pueda acudir algún especialista en la salud para que aborde ese factor de riesgo. Se considera óptimo evitar fumar,

Contacto

Correo electrónico:
 mrodriguez.6789@unicesmag.edu.co,
 cadelgado.9770@unicesmag.edu.co
 Teléfono: 3219671739, 3157155716

Figura 21. Instrucciones

Selección de datos

El aplicativo web implementa un formulario que permite recolectar los datos relevantes para la predicción de enfermedades cardiovasculares.

Análisis: requerimientos funcionales para la evaluación de riesgo cardiovascular.

<i>RF-02</i>	Ingresar los datos
<i>Versión:</i>	1.0 (01-04-2024)
<i>Autores:</i>	Michael Rafael Rodríguez Rodríguez Claudia Alejandra Delgado Calpa
<i>Descripción:</i>	La aplicación web debe permitir recolectar los datos de aquellos factores de riesgo incluidas en las medidas antropométricas
<i>Importancia:</i>	Esencial
<i>Urgencia:</i>	Inmediata
<i>Estabilidad:</i>	Alta
<i>Estado:</i>	Implementado
<i>Comentarios:</i>	Ninguno

Figura 22. Requerimiento funcional rf-02

Desarrollo

El aplicativo web cuenta con un formulario interactivo que le permite al usuario ingresar los datos antropométricos y factores de riesgo cardiovascular. Estos datos son necesarios para realizar una evaluación precisa del riesgo de enfermedades cardiovasculares.

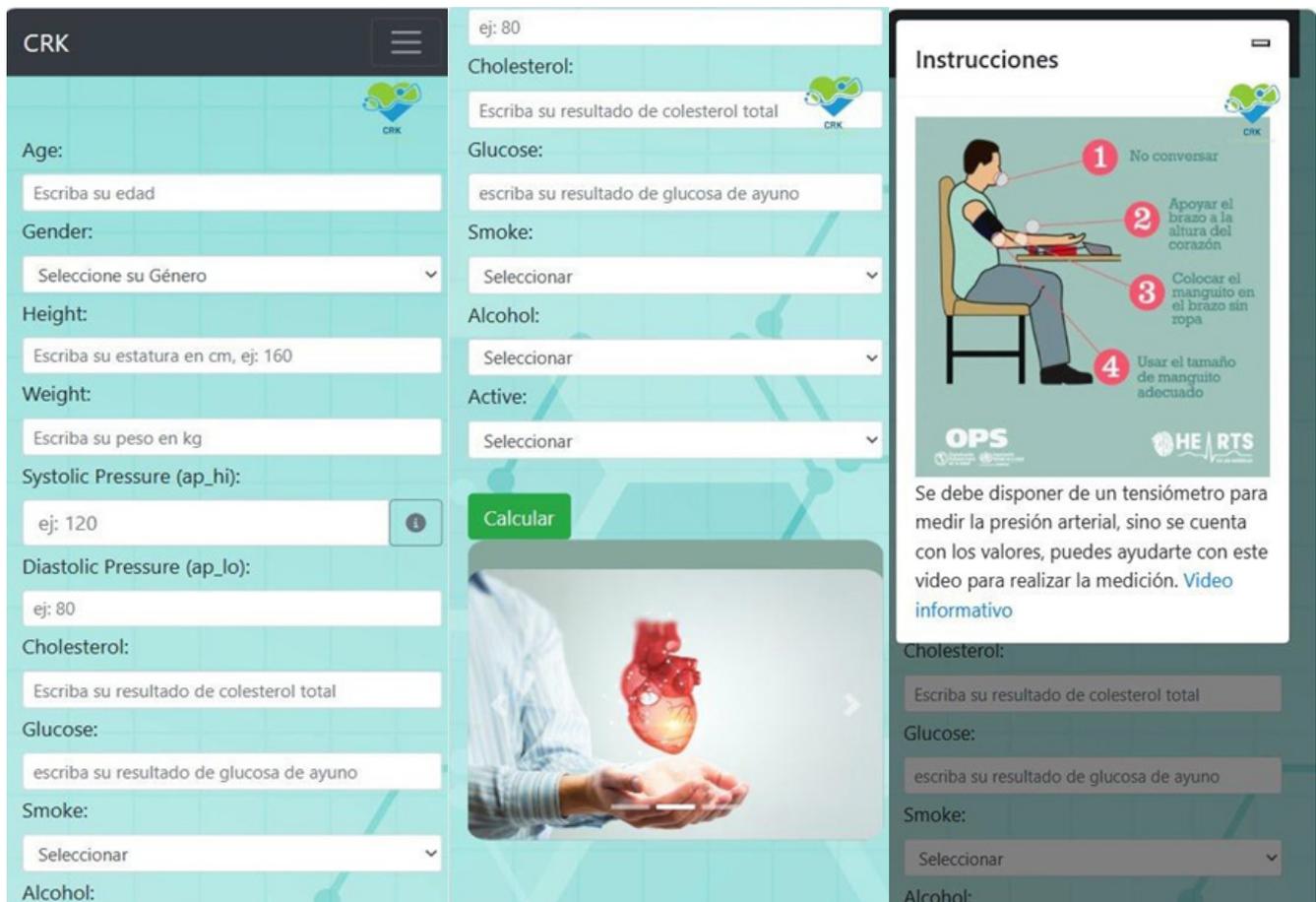


Figura 23. Formulario de datos

Visualización del Riesgo Cardiovascular (RCV)

En esta etapa el usuario podrá visualizar los resultados que se obtuvieron al complementar el formulario y el porcentaje de padecer alguna enfermedad cardiovascular (ECV).

Análisis: Requerimientos funcionales para la visualización de ECV

<i>RF-03</i>	Visualizar el porcentaje de RCV
<i>Versión:</i>	1.0 (01-04-2023)
<i>Autores:</i>	Michael Rafael Rodríguez Rodríguez Claudia Alejandra Delgado Calpa
<i>Descripción:</i>	La aplicación web debe indicar al usuario el riesgo de presentar alguna ECV con su porcentaje.
<i>Importancia:</i>	Esencial
<i>Urgencia:</i>	Inmediata
<i>Estabilidad:</i>	Alta
<i>Estado:</i>	Implementado
<i>Comentarios:</i>	Ninguno

Figura 24. Requerimiento funcional rf-03

Desarrollo

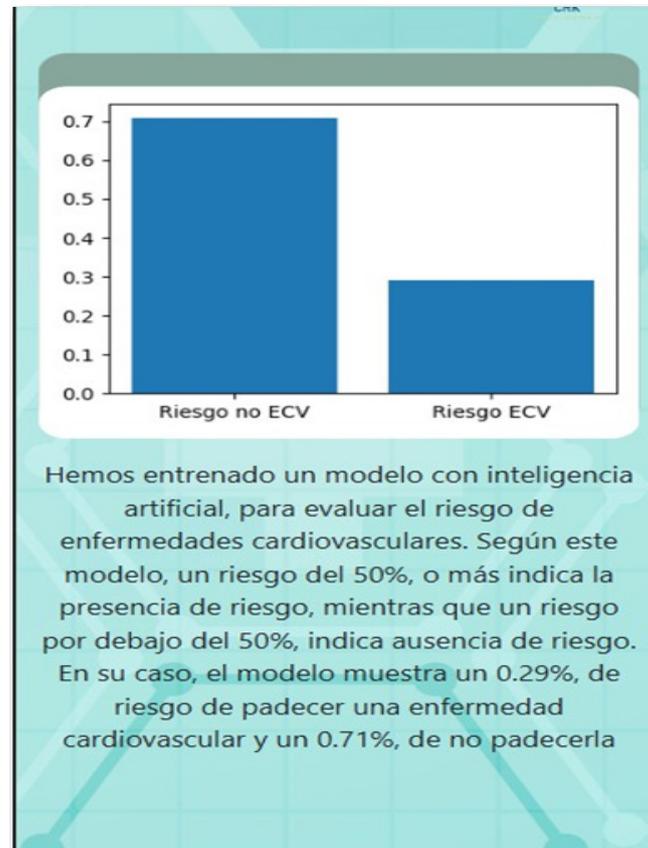


Figura 25. Porcentaje de RCV

En este caso se evalúan los datos recopilados del formulario, se visualiza el porcentaje de riesgo de padecer alguna ECV y además se podrá visualizar por medio de una gráfica el resultado.

Para ello se desarrolló un modelo avanzado para evaluar el riesgo de enfermedades cardiovasculares, utilizando una base de datos de 10 000 registros de usuarios. Este modelo fue entrenado con un enfoque binario, donde un resultado de 1 indica la presencia de riesgo y un resultado de 0 indica ausencia de riesgo. Durante el proceso de entrenamiento, el modelo aprendió a identificar patrones y factores de riesgo asociados con enfermedades cardiovasculares basándose en los datos proporcionados.

El funcionamiento del modelo se basa en la generación de un porcentaje de riesgo. Si este porcentaje supera el umbral del 50 %, el modelo clasifica al individuo como en riesgo de padecer una enfermedad cardiovascular. Por el contrario, si el porcentaje es inferior al 50 %, el modelo determina que el individuo no está en riesgo. Este método permite una evaluación clara y directa del riesgo cardiovascular, facilitando la toma de decisiones informadas sobre la salud del usuario.

Este aplicativo ha sido diseñado para ser intuitivo y accesible, permitiendo a los usuarios comprender fácilmente su nivel de riesgo sin necesidad de conocimientos técnicos avanzados. La información proporcionada por el modelo puede ser fundamental para la prevención y el tratamiento temprano de enfermedades cardiovasculares, mejorando así la salud y el bienestar de los usuarios.

Interpretación de los factores de riesgo generando alertas tempranas

Esta sección va orientada a evaluar cada factor de riesgo, permitiendo generar una alerta temprana con información útil de evaluación.

Análisis: Requerimientos funcionales para la visualización de alertas tempranas

<i>RF-04</i>	Visualizar de alertas tempranas
<i>Versión:</i>	1.0 (01-04-2023)
<i>Autores:</i>	Michael Rafael Rodríguez Rodríguez Claudia Alejandra Delgado Calpa
<i>Descripción:</i>	La aplicación web debe permitir la visualización de las alertas presentadas en cada factor de riesgo, con recomendaciones útiles para su implementación.
<i>Importancia:</i>	Esencial
<i>Urgencia:</i>	Inmediata
<i>Estabilidad:</i>	Alta
<i>Estado:</i>	Implementado
<i>Comentarios:</i>	Ninguno

Figura 26. Requerimiento funcional rf-04

Desarrollo

Las alertas tempranas se generan cuando existan variables del formulario alteradas que permitan de cierta manera generar estas alertas con recomendaciones útiles para los usuarios.



Figura 27. Alertas tempranas

Indicaciones prácticas para llevar un estilo de vida saludable

Este módulo contiene consejos prácticos adaptados por recomendaciones de expertos para llevar un estilo de vida saludable.

Análisis: Requerimientos funcionales de indicaciones prácticas

	Claudia Alejandra Delgado Calpa
<i>Descripción:</i>	La aplicación web debe proporcionar indicaciones prácticas para la prevención de ECV
<i>Importancia:</i>	Esencial
<i>Urgencia:</i>	Inmediata
<i>Estabilidad:</i>	Alta
<i>Estado:</i>	Implementado
<i>Comentarios:</i>	Ninguno

RF-05	Módulo de sugerencias prácticas
Versión:	1.0 (01-04-2023)
Autores:	Michael Rafael Rodríguez Rodríguez

Figura 28. Requerimiento funcional rf-05

Desarrollo

El módulo de Indicaciones prácticas tiene como objetivo proporcionar a los usuarios información detallada y consejos prácticos sobre cómo adoptar y mantener un estilo de vida saludable. Esto incluye recomendaciones sobre nutrición, actividad física, descanso, manejo del estrés, abandono de hábitos nocivos como el tabaquismo y el consumo excesivo de alcohol, así como pautas generales para la prevención de enfermedades y la promoción del bienestar general.



Figura 29. Recomendaciones

Requerimientos no funcionales

Debido a que la aplicación es útil para ejecutarla en diferentes escenarios, es importante tener un buen proceso de ejecución con su interfaz gráfica visible, impidiendo que esta no se adapte al dispositivo generando al usuario una mala experiencia.

<i>RNF-01</i>	Usabilidad
<i>Versión:</i>	1.0 (01-04-2023)
<i>Autores:</i>	Michael Rafael Rodríguez Rodríguez Claudia Alejandra Delgado Calpa
<i>Descripción:</i>	La aplicación debe permitir ser ejecutada en distintos dispositivos.
<i>Importancia:</i>	Esencial
<i>Urgencia:</i>	Inmediata
<i>Estabilidad:</i>	Alta
<i>Estado:</i>	Implementado
<i>Comentarios:</i>	Ninguno

Figura 30. Requerimiento no funcional rnf-01

<i>RNF-02</i>	Eficiencia
<i>Versión:</i>	1.0 (01-04-2023)
<i>Autores:</i>	Michael Rafael Rodríguez Rodríguez Claudia Alejandra Delgado Calpa
<i>Descripción:</i>	La aplicación debe tener tiempos de ejecución en el cálculo de RCV no mayor a 5 segundos en promedio.
<i>Importancia:</i>	Esencial
<i>Urgencia:</i>	Inmediata
<i>Estabilidad:</i>	Alta
<i>Estado:</i>	Implementado
<i>Comentarios:</i>	Ninguno

Figura 31. Requerimiento no funcional rnf-02

Análisis de resultados

En esta sección se llevaron a cabo diferentes entrenamientos con diferente cantidad de datos, los entrenamientos realizados fueron una parte fundamental de la investigación, puesto que se sabe que no hay un kernel superior a los demás, sino que cada uno tiene sus virtudes y comportamientos dependiendo del conjunto de datos al que se esté aplicando y su uso con cualquier algoritmo de aprendizaje automático, es importante destacar que para este caso se usó un algoritmo de clasificación. Seguido a esto, se creó el modelo con el Kernel que mejor exactitud obtuvo y que fue usado con un conjunto de datos. Finalmente, la evaluación fue realizada con el modelo y con los mejores parámetros.

En el análisis de los datos que se obtuvieron durante la experimentación, se utilizaron métricas que permitieron medir el tiempo necesario para proporcionar resultados, para ello se utilizó tiempo de entrenamiento (Training Time), esta métrica mide el tiempo total requerido para entrenar el modelo desde el inicio hasta que se completa el entrenamiento. Incluye el tiempo necesario para procesar los datos, ajustar los parámetros del modelo y realizar las iteraciones necesarias, se observó que las redes neuronales artificiales (ANN) tomaron menos tiempo que las máquinas de soporte vectorial (SVM), para todos los Kernel, debido a la aproximación de Nystroem, proporcionando los resultados como se muestra en la figura 32 y en la figura 33.

```

start_time = time.time()

annc=KANNC(kernel='can',gamma=206.913808111479,random_state=0)
modelo_predic2 = annc.fit(Xtrain, ytrain)

end_time = time.time()

training_time = end_time - start_time

print('clfk',annc.score(Xtest,ytest))
print(f'Tiempo de entrenamiento: {training_time:.2f} segundos')

clfk 0.69
Tiempo de entrenamiento: 3.67 segundos

```

Figura 32. Tiempo en ANN

```

start_time = time.time()

svc=KSVC(kernel='can', gamma=0.0006951927961775605,C=2154.4346,probability=True)
modelo_predic2 = svc.fit(Xtrain, ytrain)

end_time = time.time()

training_time = end_time - start_time

print('cmlfc',svc.score(Xtest,ytest))
print(f'Tiempo de entrenamiento: {training_time:.2f} segundos')

cmlfc 0.725
Tiempo de entrenamiento: 11.65 segundos

```

Figura 33. Tiempo de SVM

Por consiguiente, es importante destacar el acoplamiento de los modelos a la biblioteca Scikit-learn. Esto permitió ahorrar tiempo en la construcción de los algoritmos de búsqueda de hiperparámetros, logrando así un aporte a la comunidad de software basada en la construcción de modelos basados en funciones Kernel.

De acuerdo con lo enunciado por los expertos Mora.H., y Belanche.L., las funciones Kernel como la Canberra, Truncated, y adicionalmente Coseno proporcionan mejores resultados que la RBF para cierto tipo de datos, confirmando esta hipótesis en lo que respecta a esta investigación y para comprobar lo expuesto por los expertos se puede observar los resultados en la figura 3, donde la creación de esos nuevos kernel se destacan por la obtención de mejores resultados para SVM, que, para ANN, e independiente de la configuración de los mejores parámetros, comprobando que el kernel Canberra obtuvo un 72 % de rendimiento demostrando ser la opción más eficiente, frente a otros Kernels.

De esta experimentación se puede destacar que según la configuración de los parámetros se puede aproximar, a lograr mejores resultados, logrando varias comparaciones, incluyendo la de Kernels más escalables, la de algoritmos y la de transformadores de escalación. En el caso de las ANN, la comparación de transformadores de escalación en el Standar Scaler con el Kernel Canberra mostró un rendimiento del 71 % con el Kernel Radial Basic, 72 %, en cuanto al Migmag Scaler con el Kernel Canberra, se obtuvo un porcentaje del 73 % y con el Normalizer Scaler se obtuvo un porcentaje del 71 % con el Kernel Triangular.

Para las SVM, el rendimiento con el Standar Scaler el Kernel Truncated fue del 71 %, mientras que con Migmag Scaler el Kernel Truncated fue del 71 %. con el transformador Normalizer Scaler el Kernel Radial Basic, el rendimiento fue del 72 %. Se observó una similitud en los porcentajes en estas configuraciones como se muestra en la figura 3. Sin embargo, en las SVM, el Kernel Canberra demostró ser la opción más eficiente.

Concluyendo así que las ANN mostraron un rendimiento inferior al 50 %, mientras que las SVM superaron el 70 %. Esto indica que es más favorable utilizar algoritmos de clasificación basados en SVM. Una de las características de los Kernels que permitió esta conclusión fue el estudio de la exactitud en Máquinas de

Soporte Vectorial donde se muestra cada Kernel con su exactitud observando que un modelo con valores de exactitud mayores al 50 % se clasifica en aceptable lo que permite el desarrollo de un buen modelo, y empeoran para ANN porque como se muestra, cada Kernel con su exactitud muestra valores que están por debajo del 50 % en su mayoría, por lo tanto un modelo con menos del 50 % de exactitud se considera muy pobre esto significa que el modelo está prediciendo incorrectamente más de la mitad del tiempo.

Dentro del marco de este proyecto centrado en la comparación de funciones kernel en SVM y ANN en la predicción del riesgo cardiovascular, se ve la necesidad de que ese modelo sea desplegado mediante el aplicativo. Para ello este entorno de investigación abarca poblaciones como individuos con factores de riesgo, personas con antecedentes cardiovasculares y aquellos que podrían beneficiarse de intervenciones preventivas sin padecer alguna enfermedad cardiovascular. Además, se considera la calidad de la atención médica, la disponibilidad de tecnologías de monitoreo y las barreras que enfrentan las personas en la detección temprana y prevención de eventos cardiovasculares. El objetivo es mejorar la salud cardiovascular a través de la aplicación de modelos de inteligencia artificial y el análisis de datos para la predicción temprana y la implementación de estrategias preventivas.

La herramienta gráfica desarrollada en este proyecto se basa en la visualización interactiva de datos relacionados con el riesgo cardiovascular. Esta herramienta fue implementada utilizando tecnologías avanzadas de visualización de datos como Python, integrándose con modelos de inteligencia artificial que procesan y analizan datos clínicos de diversas fuentes. La implementación incluyó las siguientes etapas:

1. Se recopilaron datos de diferentes fuentes. Estos datos fueron preprocesados para garantizar su calidad y coherencia.
2. Utilizando técnicas de Machine Learning, se desarrolló un modelo predictivo para evaluar el riesgo cardiovascular de los usuarios. Estos modelos fueron entrenados y validados utilizando conjuntos de datos relevantes.
3. La herramienta gráfica se integró con el modelo predictivo para permitir una visualización intuitiva de los resultados. Se desarrollaron módulos interactivos que permiten a los usuarios explorar los datos y entender mejor los factores de riesgo y las predicciones del modelo.

La visualización gráfica de los datos permite a los profesionales de la salud tomar decisiones informadas basadas en la comprensión clara de los riesgos individuales y colectivos, siendo esta herramienta un apoyo al diagnóstico médico. Esto es particularmente útil para identificar patrones y tendencias que no son fácilmente discernibles a través de datos tabulares, a su vez la herramienta ayuda a identificar a individuos en alto riesgo de desarrollar enfermedades cardiovasculares antes de que se presenten síntomas clínicos evidentes. Esto facilita la implementación de estrategias preventivas y de intervención temprana, mejorando los resultados de salud a largo plazo.

La interfaz gráfica está diseñada para ser intuitiva y accesible, incluso para usuarios con poca experiencia en análisis de datos. Esto permite a los usuarios y a los profesionales de la salud entender mejor la información presentada y tomar decisiones basadas en datos. Además, sirve como una herramienta educativa que puede ilustrar claramente los efectos de diferentes factores de riesgo y las ventajas de las intervenciones preventivas.

CONCLUSIONES

El escenario experimental propuesto permitió confirmar lo postulado por los expertos Mora.H., y Belanche.L., y validar el proyecto de investigación de que las principales características de los kernel investigados se ajustan y se comportan mejor ante el tipo de datos suministrados para Maquinas de Soporte Vectorial (SVM) y empeoran para Redes Neuronales Artificiales (ANN).

El acoplamiento de las funciones kernel a la biblioteca Scikit-Learn facilita una relación cercana con algoritmos desarrollados, contribuyendo así a la comunidad de software libre e investigación académica. Esto sirve como punto de referencia para construir modelos que se ajusten a esta biblioteca que es utilizada globalmente.

La visualización de los resultados de los experimentos permitió evaluar objetivamente el rendimiento de cada Kernel. Esto nos permitió identificar los mejores parámetros para las máquinas de soporte vectorial, utilizando normalizadores de escalación como Standar Scaler y Kernel Truncated para la clasificación de datos de riesgo cardiovascular, logrando un buen equilibrio entre tiempo de procesamiento y exactitud.

Finalmente, el desarrollo de la herramienta permitió desplegar de manera gráfica el modelo y facilito el aprendizaje en la implementación de interfaces gráficas dinámicas, de usuario en Python, además del uso de bibliotecas de análisis de datos disponibles en Python.

La herramienta desarrollada es fácilmente escalable, se recomienda su utilización para implementar nuevos modelos y funciones. Además, es importante desarrollar nuevas aplicaciones de visualización que faciliten la predicción de enfermedades cardiovasculares.

Dado que la investigación incluye un comparativo de funciones kernel tanto en Máquinas de Soporte Vectorial

(SVM) como en Redes Neuronales Artificiales (ANN), donde fue necesario profundizar en la teoría de métricas utilizadas y matemáticas, se considera necesario promover entre los estudiantes la importancia de comprender los fundamentos de estos principios para abordar las bases de cualquier tema de investigación.

Por último, se recomienda profundizar en los estudios comparativos de funciones kernel acopladas a otros algoritmos de Machine Learning para clasificación o regresión lineal de predicción de enfermedades cardiovasculares, para saber qué algoritmo trabaja mejor con Kernels en estos conjuntos de datos.

REFERENCIAS BIBLIOGRÁFICAS

1. Beade Ruelas A, García Soto CE. No rompas más tu corazón. Salud cardiovascular. Gobierno de México; 2017. <https://www.gob.mx/profeco/documentos/no-rompas-mas-tu-corazon-salud-cardiovascular>

2. Cemprende. Enfermedades cardiovasculares, un gran desafío para la salud pública en Colombia. Innpulsa Colombia; 2022. <https://www.innulsacolombia.com/cemprende/noticias/enfermedades-cardiovasculares>

3. Molina de Salazar D. Propuesta en prevención del riesgo cardiovascular. *Rev Colomb Cardiol*. 1988;5(5):20888.

4. Patiño Zambrano CF. Dispositivo vestible inteligente para la generación de alertas tempranas de eventos cardiovasculares de riesgo. Envigado; 2022.

5. Sanofi Campus. Machine Learning y predicción de enfermedades cardiovasculares. 2022. <https://campus.sanofi.es/es/noticias/machine-learning-prediccion-enfermedades-cardiovasculares>

6. Gómez LA. Las enfermedades cardiovasculares: un problema de salud pública y un reto global. *SciELO*. 2011;1.

7. Pérez Leal LE, Buitrago Cárdenas JA. Predicción del diagnóstico de diabetes a partir de perfiles clínicos de pacientes utilizando aprendizaje automático. Bogotá: Universidad Antonio Nariño; 2021.

8. Gallego Valcárcel DA, Lucas Monsalve DF. Modelos de aprendizaje automático para la predicción del riesgo de fatalidad por insuficiencia cardíaca con datos clínicos. Bogotá: Universidad Antonio Nariño; 2021.

9. Álvarez Vega M, Quirós Mora LM, Cortés Badilla MV. Inteligencia artificial y aprendizaje automático en medicina. *Rev Méd Sinergia*. 2020;5(8):12.

10. Ferreira Guerrero DP, Díaz Vera M, Bonilla Ibáñez CP. Factores de riesgo cardiovascular modificables en adolescentes escolarizados de Ibagué 2013. Repositorio Institucional Universidad de Antioquia. 2017;1.

11. Mora Paz HA. Comparativo de Kernels sobre predicción de oferta de fuentes alternativas de energía. Pasto: UNIR La Universidad en Internet; 2019.

12. Friedman PA, Kapa S, López Jiménez F, Noseworthy PA. Inteligencia artificial en cardiología. *Mayo Clinic*; 2023. <https://www.mayoclinic.org/es-es/departments-centers/ai-cardiology/overview/ovc-20486648>

13. Aprende IA. Kernel y máquinas de vectores de soporte. Aprende IA. <https://aprendeia.com/kernel-maquinas-vectores-de-soporte-clasificacion-regresion/>

14. Sowmya V, Sanjana K, Gopalakrishnan E, Soman KP. Inteligencia artificial explicable para la variabilidad de la frecuencia cardíaca en la señal de ECG. *Health Technol Lett*. 2020;7(6):146.

15. Mohan S, Thirumalai C, Srivastava G. Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques. *IEEE Access*. 2019;7:81542-54.

16. Wu H, Yang L, Jin X, Zheng P. Study of cardiovascular disease prediction model based on random forest in eastern China. *Sci Rep*. 2020;10(1):5245.

17. Chavez Olivera O, Galindo Honores L, Barrientos Padilla A, Cuadros Galvez M. Aplicación móvil para predecir la probabilidad de pertenecer al grupo de riesgo cardiovascular utilizando machine learning. En: XII Conf Iberoamericana de Complejidad, Informática y Cibernética. Lima; 2022.

18. Scavino M, Castrillejo A, Estragó Mérola VS, Luraghi López LE, Muñoz M, Álvarez Vaz R. Informe final publicable del proyecto de creación de algoritmos utilizando técnicas de clasificación supervisada y no supervisada para el diagnóstico de enfermedades cardiovasculares. Uruguay; 2022.
19. Polero LD, Garmendia CM, Echegoyen RE, Alves de Lima A, Bertón F, Lambardi F, et al. Predicción de riesgo de sufrir un síndrome coronario agudo mediante un algoritmo de Machine Learning (ANGINA). Rev Argent Cardiol. 2020;88(1).
20. Perez Tatis JD. Optimización de un modelo de clasificación de enfermedades cardiovasculares utilizando técnicas de aprendizaje profundo supervisado y despliegue de dashboard web. Cartagena; 2021.
21. Martínez EJ. Predicción de enfermedades cardiovasculares mediante algoritmos de inteligencia artificial. Málaga; 2020.
22. Carrascal Porras FL, Florez Prias LA. Modelo de inteligencia artificial como apoyo diagnóstico para la estimación de riesgo cardiovascular en pacientes atendidos bajo la modalidad de telemedicina. Sucre: UNAD; 2021.
23. Dolores C, Ordovás J. Genes, dieta y enfermedades cardiovasculares. Genética. 2007;5:71-118.
24. Vega Abascal J, Guimará Mosqueda M, Vega Abascal L. Riesgo cardiovascular, una herramienta útil para la prevención de las enfermedades cardiovasculares. Rev Clin Invest Arterioscler. 2011;27(1):3.
25. Martínez EJ. Predicción de enfermedades cardiovasculares. Málaga: Universidad de Málaga; 2020.
26. Sans Menéndez S. Enfermedades cardiovasculares. Barcelona: Institut d'Estudis de la Salut; 2011.
27. Diabetrics. Escala Framingham. Diabetrics. <https://blog.diabetrics.com/escala-framingham>
28. Perez Tatis JD. Optimización de un modelo de clasificación de enfermedades cardiovasculares utilizando técnicas de aprendizaje profundo supervisado y despliegue de dashboard web. Cartagena: Universidad del Sinú; 2021.
29. MathWorks. Máquina de soporte vectorial (SVM). <https://es.mathworks.com/discovery/support-vector-machine.html>
30. Ecured. Función kernel. https://www.ecured.cu/Funci%C3%B3n_Kernel
31. Wikipedia. Clasificador lineal. 2019. https://es.wikipedia.org/wiki/Clasificador_lineal
32. Tibco Data Science. ¿Qué es el aprendizaje supervisado? <https://www.tibco.com/es/reference-center/what-is-supervised-learning>
33. Montiel de Jesús A. Desarrollo de una aplicación para dispositivos móviles para la detección temprana de enfermedades cardiovasculares. Orizaba, México: TECNM; 2022.
34. Sitio BigData. Modelos de Machine Learning: Métricas de regresión. 2019. <https://sitiobigdata.com/2019/05/27/modelos-de-machine-learning-metricas-de-regresion-mse-parte-2/>
35. Martínez Heras J. Métricas de clasificación: precisión, recall, F1, accuracy. IArtificial; 2020. <https://www.iartificial.net/precision-recall-f1-accuracy-en-clasificacion/>
36. Quintanilla L, Warren G, Kirsch S, Youssef V, Kershaw N, Killeen S, et al. Learn Microsoft: métricas de ML. 2023. <https://learn.microsoft.com/es-es/dotnet/machine-learning/resources/metrics>
37. Boehm B. Desarrollo en espiral. Wikipedia; 2012. https://es.wikipedia.org/wiki/Desarrollo_en_espiral
38. Ballina Ríos F. Paradigmas y perspectivas teórico-metodológicas en el estudio de la administración. UVMX; 2013.

39. Radrigán M. Método empírico-analítico. Wikipedia; 2022. https://es.wikipedia.org/wiki/M%C3%A9todo_emp%C3%ADrico-anal%C3%ADtico
40. IBM. Conceptos básicos de ayuda de CRISP-DM. 2021. <https://www.ibm.com/docs/es/spss-modeler/sas?topic=dm-crisp-help-overview>
41. Vallalta Rueda JF. CRISP-DM: una metodología para minería de datos en salud. IA Health Data Miner. <https://healthdataminer.com/data-mining/crisp-dm-una-metodologia-para-mineria-de-datos-en-salud/>
42. Toral Barrera JA. Redes neuronales. España: CUCEI; 2019.
43. Kaggle. Herramientas de repositorio. 2018.
44. Asunción A, Newman D. Repositorios. Rexa.info: Massachusetts Amherst; 2007.
45. Mora Paz H, Riascos JA, Salazar Castro JA, Mora G, Pantoja A. Comparación de funciones kernel para la predicción de la oferta energética fotovoltaica. RISTI. 2020;(E38):310-24.
46. Belanche LA, Villegas MA. Kernel functions for categorical variables with application to problems in the life sciences. 2023;(08034):1-3.
47. Esri. Spatial analysis in ArcGIS Pro. ArcGIS Pro. <https://pro.arcgis.com/es/pro-app/latest/help/analysis/introduction/spatial-analysis-in-arcgis-pro.htm>
48. Scriptología. Tutorial de Flask: desarrollando aplicaciones web en Python. 2024. <https://scriptologia.com/tutorial-de-flask-desarrollando-aplicaciones-web-en-python/>
49. Hunter J, Dale D, Firing E, Droettboom M. Introducción a pyplot. Matplotlib; 2012. <https://es.matplotlib.net/stable/tutorials/introductory/pyplot.html>
50. Python. Biblioteca Pickle. 2001. <https://docs.python.org/es/3/library/pickle.html>
51. DataScientest. Uso de pandas en Python. 2023. <https://datascientest.com/es/pandas-python>
52. Manav N. Escribir bytes a archivo en Python. 2023. <https://www.delftstack.com/es/howto/python/write-bytes-to-file-python/>
53. Python. Biblioteca base64. <https://docs.python.org/es/dev/library/base64.html>
54. Navarro S. ¿Para qué sirve el train-test split? KeepCoding; 2024. <https://keepcoding.io/blog/para-que-sirve-el-train-test-split/>
55. Scikit Learn. Nystroem Kernel Approximation. 2007. https://scikit-learn.org/stable/modules/generated/sklearn.kernel_approximation.Nystroem.html
56. Li B, Lu P, chmcl v. Multiclass Neural Network. Microsoft Learn; 2023. <https://learn.microsoft.com/es-es/azure/machine-learning/component-reference/multiclass-neural-network?view=azureml-api-2>
57. Imbert A, Lemaitre G. sklearn.svm.SVC. Scikit-Learn; 2024. <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>
58. Ruiz M. ¿Qué es Redux? OpenWebinars; 2018. <https://openwebinars.net/blog/que-es-redux/>
59. Moes T. ¿Qué es Windows? SoftwareLab; 2023. <https://softwarelab.org/es/blog/que-es-windows/>
60. D. A. Qué es Bootstrap. Hostinger; 2023. <https://www.hostinger.mx/tutoriales/que-es-bootstrap>
61. Monk R. What is Python used for. Coursera; 2023. <https://www.coursera.org/mx/articles/what-is-python-used-for-a-beginners-guide-to-using-python>

62. Valiente FT. Aprendizaje por refuerzo. IIC UAM. <https://www.iic.uam.es/inteligencia-artificial/aprendizaje-por-refuerzo>
63. Arceo Vilas AM. Estado nutricional y adherencia a la dieta mediterránea en población mayor de 40 años: IA vs estadística clásica. A Coruña: Universidad de Coruña; 2020.
64. Sánchez Santos JM, Sánchez Fernández PL. Predicción de eventos cardiovasculares y hemorrágicos en pacientes con doble antiagregación con modelos ML. CREDOS. Salamanca; 2020.
65. Gallego Valcárcel DA, Lucas Monsalve DF. Modelos de aprendizaje automático para la predicción del riesgo de fatalidad por insuficiencia cardiaca con datos clínicos. Bogotá: Universidad Antonio Nariño; 2021.
66. Fernández García P, Vallejo Seco G, Livacic-Rojas PE, Tuero Herrero E. Validez estructurada para una investigación cuasi-experimental de calidad. Rev Científica. 2014;30(2):5.
67. Lozada J. Investigación aplicada. Dialnet UNIRIOJA. 2014;3(1):47-50.
68. Londoño Ocampo M. Definición de un modelo de clasificación de riesgo cardiovascular en adultos mayores usando técnicas de aprendizaje de máquinas. Medellín: Universidad Nacional de Colombia; 2020.
69. Núñez Cárdenas FJ, Zavaleta Chi IDC, Felipe Redondo AM, Meléndez Hernández J. Aplicación de minería de datos para tipificación de ECV en alumnos universitarios. México; 2018.
70. Martínez J. Más allá del accuracy: precision, recall y F1. Datasmarts; 2019. <https://datasmarts.net/es/mas-alla-del-accuracy-precision-recall-y-f1/>
71. Mostaza JM, Pintó X, Armario P, Masana L, Real JT, Valdivielso P, et al. Estándares SEA 2022 para el control global del riesgo cardiovascular. Clin Invest Arterioscler. 2022;34(3):130-79.
72. Calvo Martín J. La importancia de las funciones de activación en una red. LinkedIn; 2022. <https://es.linkedin.com/pulse/la-importancia-de-las-funciones-activaci%C3%B3n-en-una-red-calvo-martin>

FINANCIACIÓN

Los autores no recibieron financiación para el desarrollo de la presente investigación.

CONFLICTO DE INTERESES

Los autores declaran que no existe conflicto de intereses.

CONTRIBUCIÓN DE AUTORÍA

Conceptualización: Michael Rafael Rodríguez Rodríguez, Claudia Alejandra Delgado Calpa, Héctor Andrés Mora Paz.

Curación de datos: Michael Rafael Rodríguez Rodríguez, Claudia Alejandra Delgado Calpa, Héctor Andrés Mora Paz.

Análisis formal: Michael Rafael Rodríguez Rodríguez, Claudia Alejandra Delgado Calpa, Héctor Andrés Mora Paz.

Redacción - borrador original: Michael Rafael Rodríguez Rodríguez, Claudia Alejandra Delgado Calpa, Héctor Andrés Mora Paz.

Redacción - revisión y edición: Michael Rafael Rodríguez Rodríguez, Claudia Alejandra Delgado Calpa, Héctor Andrés Mora Paz.