EthAIca. 2025; 4:172
doi: 10.56294/ai2025172

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 Rodríguez[1], Claudia Alejandra Delgado Calpa[1], Héctor Andrés Mora Paz[1]

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

Cite as: 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). EthAIca. 2025; 4:172. https://doi.org/10.56294/ai2025172

Submitted: 16-06-2024 **Revised:** 08-11-2024 **Accepted:** 03-07-2025 **Published:** 04-07-2025

Editor: PhD. Rubén González Vallejo

Corresponding Author: 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 Maquinas de Soporte Vectorial (SVM), haciendo un comparativos 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.

INTRODUCTION

Cardiovascular diseases are disorders of the heart and blood vessels, and have become the leading cause of death worldwide.[1,2,3,4,5,6,7,8,9,10,11] In Colombia, during 2021, a total of 51 988 deaths were registered as a result of CVD-related conditions, 12 % more than in 2020, with a significant increase in the incidence of cases in women, according to figures from the National Department of Statistics (DANE) 4. Although the country has made progress in efforts to reduce the burden of these scenarios, fragmented actions are still present within the health system, and the individual sacrifices of each actor are insufficient.[2,6,12,13,14,15,16,17,18,19,20] Various elements may contribute to inefficiency in the timely prevention of these pathologies, including inadequate care in health centers, limited access to technologies, delayed diagnosis, and scarce information on modifiable factors that can directly affect their health, potentially leading to severe complications in the future.[20,21,22,23,24,25,26,27,28,29,30]

For these reasons, health professionals are forced to change the way they approach patients' conditions from the principles of health promotion and disease prevention as a new public health policy.[2,3,31,32,33,34,35,36,37,38,39,40]

In other words, as mentioned by Arias[4], even if one manages to ensure access to a given health service, especially in the most vulnerable sectors, the benefit is null if one cannot save lives and improve them with the resources allocated for these diseases, in which a monitoring and analysis outside the clinical setting continues to be fundamental to timely discover the associated complications in a large part of the population. Rapid action at any location is vital; it is essential to reduce the time to detection and care to minimize the chances of cardiovascular damage, coma, or, in the worst-case scenario, death.[41,42,43,44,45,46,47,48,49,50,51]

Some technologies promise to revolutionize the field of cardiovascular disease prevention. Likewise, thanks to their incredible capabilities, they are the perfect tool to improve their prediction.

Consequently, various studies have been conducted, whose results indicate that machine learning can significantly aid in identifying cardiovascular episodes or diseases in healthy individuals.[5,52,53,54,55,56,57,58,59,60]

The following research aims to determine from the algebraic and transcendental functions, the kernel function that offers the best compromise in the prediction of cardiovascular diseases in SVM and ANN algorithms, providing researchers with kernel functions, from a fair comparative study, where the accuracy and prediction time of each function is measured; Afterwards, these new kernel functions are coupled to the Scikit-learn library, the configuration is validated by taking the most appropriate for the prevention of CVD taking into account the error-time relationship, and thus managing to build the mobile application for early warnings by implementing the predictive model.[61,62,63,64,65,66,67,68,69,70,71,72]

From a set of transcendent functions, which kernel function has the best compromise between accuracy and prediction time, to support the diagnosis of cardiovascular diseases?

General Objective

To determine from the algebraic and transcendental functions, the kernel function that offers the best compromise in the prediction of cardiovascular diseases in the SVM and ANN algorithms from a fair comparative study, where the accuracy and prediction time of each kernel function are measured.

METHOD

Paradigm

This research is framed within the positivist paradigm, a paradigm that is rational, objective, and based on verifying facts and particularities of scientific knowledge, characteristic of the engineering discipline's essence, which is in turn characterized by a high interest in verifying knowledge through predictions. Some call it the "predictionist paradigm" because the important thing is to pose a series of hypotheses, such as predicting that something will happen, and then verify or test them. It is in the exact and natural sciences where it has the most excellent application.[38] The present project enables the study of SVM and ANN algorithms through the use of kernel functions, performing systematic hyperparameter tuning configurations, and applying these methods to the data. Furthermore, the presence of variables in the datasets significantly influences further prediction and subsequent analysis.

Approach

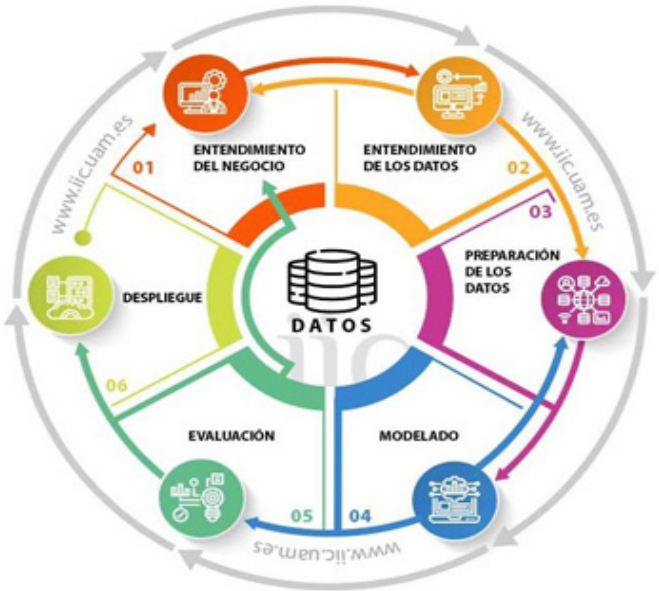
The approach to be used in this research is empirical-analytical, it is a method of observation used to deepen the study of phenomena, being able to establish general laws from the connection that exists between cause and effect in a given context, it is also one of the models to describe the scientific method, which is based on experimentation and empirical logic.[39] It addresses the reality of observable, quantifiable, and measurable facts. It is a method that rigorously contrasts its hypotheses through scientific demonstration, determining whether a hypothesis is true or false.[38] The current project enables the observation and analysis of the causes and effects of the studied problem, proposing experimentation and testing processes that will serve as a basis for proposing a solution. In this way, appropriate models will be developed for predicting cardiovascular diseases.

Method

The method will be quasi-experimental, as its objective is to test a causal hypothesis by manipulating an independent variable and then analyzing the consequences of the manipulation on one or more dependent variables.

This will be reflected in the support for clinical diagnosis in the prediction of cardiovascular diseases, promoting, in turn, hyperparameters for obtaining predictions in the SVM and ANN algorithms against the dataset of cardiovascular disease factors.

The methodology to be used in the project is CRISP-DM, which stands for Cross-Industry Standard Process for Data Mining, a proven method to guide data mining work. As a methodology, it includes descriptions of the typical phases of a project, the tasks required in each phase, and an explanation of the relationships between the functions.[40]



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

Figure 1. Schematic diagram of the standard CRISP-DM cycle

The CRISP-DM methodology establishes a data mining project as a sequence of phases:[41]

1. Business Understanding: the objective of this phase is to align the data mining project's goals with the business objectives.
2. Data compression: two key points in this phase are understanding the data, its structure, and distribution, as well as ensuring data quality.
3. Data preparation: the ultimate goal of this phase is to obtain the final data on which the models will be applied. The ultimate goal of this phase is to develop a model that enables us to achieve the project objectives.
4. Modeling: the ultimate goal of this phase is to develop a model that enables us to achieve the project objectives.
5. Evaluation: in this phase, we focus on evaluating how closely the model aligns with the project objectives.
6. Deployment: the ultimate goal of this phase is to deploy the results obtained so that they can be propagated to end users, as well as to maintain them once the deployment has been completed.

At the moment the projects that have to do with the development of intelligent models are focused on two specific methodologies, one is KDD and the other is CRISP-DM, where the latter is a more dynamic methodology, which allows the interaction of the model cycle and also of the software, achieving an interactive development, thus allowing the model to be scalable, it has that dynamism of feedback that has some phases that make up the methodology.

Type of research

The type of research developed throughout the project is determined by the kind of applied research being conducted. It pursues the creation of knowledge by applying it directly to social problems. It is based on the technological results of basic research, which deals with the relationship between concept and product.

Research design

The research design is testing because different types of tests will be performed, such as analysis in terms of performance, confidence by training algorithms giving a prediction, the project is based on two tools that takes into account the process stages of support vector machines and artificial neural networks, in this way the following phases are proposed.

- Connection to the data: it is mainly necessary to have a repository that provides us with data from different authors, thus allowing an estimation of the data obtained to be sufficient and relevant for model building.
- Definition of an evaluation criterion: this is a measure of error, commonly the use of the mean square error that allows its use in measures such as accuracy and completeness.
- Preparation of the data: complete data, combination of data from various sources, appropriate formatting of the data, keeping in mind the relevant characteristics, and subsequently the construction of the model.
- Error analysis: use different SVM and ANN models, check the data if it is required in its entirety or only some of it is taken.
- Training phase: when the type of artificial neuron to be used in a neural network is selected and the topology is determined, the training process of the network begins so that it can be used. From a set of random synaptic weights, the learning process searches for a set of weights that allow the network to correctly perform a specific function.
- Operation phase: once the learning phase is finished, the model may become too adjusted to the particularities present in the training patterns, losing its ability to generalize its learning to new cases.[42]
- Analysis of results and knowledge discovery: the results of the methods will be analyzed through visualizations, so that it can determine which methods performed better on the different datasets, this allows researchers to test their methods in SVM, ANN and thus allows a safe prediction of cardiovascular diseases and this tool allows to be used as diagnostic aids.

Population

The purpose of this research is to study the epistemological and mathematical part of the object of study, which in this case are the kernel functions applied to SVM and ANN, therefore, neither population nor samples are required, as mentioned in the maximum distinction research Evaluation of Dimension Reduction Methods For The Topological Preservation Of Data Using Rnx Metrics developed by (Correa Lozano. C.D., Lozano Thomé.J.A., Urrea Burgos.D.F.), since the nature of the project demands a Data Driven Approach type of research, in which the databases are the main ones in the development of the project. The Datasets that will be worked on will be tested on different datasets. This project will not use focal data but will utilize various databases already established by other researchers.

Data collection techniques

This project does not require the collection of new information, as databases with standard characteristics are already available and used by different researchers to test various methods of cardiovascular risk prediction. But the databases used are listed below and will be taken into account:

Kaggle dataset repository tools

A national institute responsible for education evaluation, to facilitate the use and consultation of the test results databases, has made the digital FTP available to the academic community for researchers. This tool serves as a repository for research groups and advanced-level students interested in participating in research calls, providing them with firsthand information to develop their proposals. From the crossing of available databases, it is possible to compile data that allows for the classification of students into different socioeconomic levels. It is of interest in this project to predict a student's socioeconomic level from the information in the database.

To do this, use all the prediction techniques seen (some to be seen) in class:

- Knn - Nearest neighbors.
- Classification trees
- Neural Networks.
- Random Forest
- Boosting, Gradient Boosting
- LDA, QDA, FDA.[43]

UCI dataset repository tools

The UCI machine learning repository is a collection of databases, domain theories, and data generators used by the machine learning community for empirical analysis of machine learning algorithms. The archive

was created as an ftp file in 1987 by David Aha and his fellow graduate students at UC Irvine. Since then, it has been widely used by students, educators, and researchers worldwide as a primary source of machine learning datasets. As an indication of the impact of the archive, it has been cited over 1000 times, making it one of the top 100 most cited "articles" in all of computer science. The current version of the website was designed in 2007 by Arthur Asuncion and David Newman, and this project is in collaboration with Rexa.info at the University of Massachusetts Amherst. Financial support from the National Science Foundation is gratefully acknowledged.[44]

Validity of the technique

The variables in the repository dataset will be preprocessed using cleaning algorithms, scaling transforms, and dimension reduction algorithms. To store the results of each experiment, especially for the kernel functions, they will be stored in a dataframe whose data will be saved in a CSV file.

Reliability of the technique

This project, as mentioned in previous sections, adopts machine learning techniques to generate data that can significantly reduce the potential margin of error. This will be achieved by adjusting the algorithm settings appropriately. Additionally, the results obtained from these tests will be used to validate or refute the hypothesis presented.

RESULTS

This chapter presents the results of the research that correspond to the development of each of the stated objectives.

Definition of Kernel functions recommended by experts

Results for the definition of kernel functions with expert recommendation

To execute the result of the first objective, which is to define Kernel functions based on the recommendation of experts, we proceeded to the search of two essential documents having a different subject matter between them, and at the same time a close relationship with the use of Kernel functions, their behavior with a variety of data and which are the most adequate to fit the created model. It begins with a review of the scientific literature related to the use of Kernel functions, citing these experts, which shows that there is a theoretical basis for exploring alternative Kernel functions.

One of the articles of which is taken into account for the development of this research is the article of Mora.H., in his project entitled "Comparison of Kernel functions for the prediction of photovoltaic energy supply" published in the year 2020, Spain[45], and the other expert Belanche.L., in his project entitled "Kernel Functions for Categorical Variables with Application to Problems in the Life Sciences" published in the year 2013, Barcelona, Spain[46], it can be evidenced that they make use of alternative Kernel functions, generally in the implementations of the libraries that are in machine learning and that in turn 4 algorithms of Linear, Polynomial, RBF, Sigmoid Kernel functions are implemented, taking the latter Belanche.L that with his project presents that the proposed approach can remarkably outperform the Standard Kernels, so it can be used as a good alternative to other standard Kernel functions (at least for SVM classification) to obtain better accuracy, with alternative functions can be acquired better results than the RBF, since it is highlighted that the Categorical Kernel takes advantage of information that is not used by the other Kernels, such as$K$ 0 or the RBF Kernel. This suggests that the Categorical Kernel can capture relationships and patterns in the data that the other Kernels cannot capture.

It is also important to note that the paper mentions that in cases where the results of the Standard Kernels are good, the Categorical Kernel K1 improves slightly compared to K0, and both outperform the RBF Kernel. This indicates that the proposed approach is beneficial even when the results are already acceptable. It is noted that the RBF Kernel exhibits poor performance in some instances, likely due to its susceptibility to overfitting on small sample data in high-dimensional representations.

And the first mentioned paper by Mora, H., uses a pair of Kernel functions: the Gaussian and Polynomial. From this research, it is highlighted that alternative Kernels have only improved results for SVM and worsened for ANN. It is hypothesized that the Kernels work better for SVM than in other linear classification and prediction algorithms because this algorithm uses the maximum margin criterion, opening the discussion to perform new experiments that take into account more machine learning algorithms that use Kernel functions, evaluating these algorithms with different databases for classification and prediction. The results of the analysis indicate that SVM with the Standard and Min-Max normalizers, and the Triangular and Rational Quadratic Kernels, are the most suitable for regression on Landsat and MODIS data. At the same time, the Kernel functions proposed by both Belanche and Mora are those defined in the following table.

Kernel	
Gaussiano (RBF)	**Triangular (tr)**
$\mathbf{e}^{-\sum_{i}^{d}=i\,\gamma(x_i-x_i')^{\beta}}$ $\boldsymbol{\gamma > 0, \beta \epsilon (0,2]}$	$\begin{cases} \|\|x-x'\\| \le a \to 1-\frac{\|\|x-x}{a} \\ \|\|x-x'\\| > a \to 0 \end{cases}$ $a > 0$
Radial Basic (Rb)	**Rational Quadratic (Rq)**
$(\sum_{i=1}^{d} \mathbf{e} = \boldsymbol{\gamma}(\boldsymbol{x_i} - \boldsymbol{x_i}')^{\mathbf{2}})^{m}$ $\boldsymbol{\gamma > 0, m \epsilon N}$	$1-\frac{\|\|x-x'\\|^2}{\|\|x-x'\\|^2+a}$ $a > 0$

Canberra (Can)	Truncated (Tru)
$\mathbf{1}-\frac{\mathbf{1}}{\mathbf{d}}\sum_{\mathbf{i}}^{\mathbf{d}} \boldsymbol{\gamma}\frac{\|\mathbf{x_i}-\mathbf{x_i}'\|}{\|\mathbf{x_i}\|+\|\overline{\mathbf{x_i}'}\|}$ $\boldsymbol{\gamma \epsilon (0,1]}$	$\frac{1}{d}\sum_{i}^{d} \max(0, \frac{\|x_i-x'\|}{\gamma})$ $\gamma > 0$

Source: comparison of kernel functions on prediction of supply of alternative energy sources
Figure 2. Mathematical formulation of the kernel functions used by experts

And of which additionally based on belanche's definition:

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

From which the following kernels are proposed as exploration:

A Radial Logarithmic Basic kernel: this is a similarity function used to measure the distance between data points in a feature space. The RBF kernel can map data to a higher dimensionality feature space where the data can be more easily separated by a hyperplane. This allows algorithms such as SVMs to perform classification and regression on data sets that are not linearly separable in their original space. Since the RBF kernel can capture nonlinear patterns, it is useful in situations where the data has complex, nonlinear relationships.[47]

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

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

All this above analysis led us to define Kernels inspired by the conception of alternative activation functions in neural networks and they are:

Kernel	
Polynomial (pk)	**Triangular (tk)**
$\boldsymbol{K(x,x') = (\|x,x'\| + c)^d}$	$\begin{cases} \|\|x-x'\\| \le a \to 1-\frac{\|\|x-x}{a} \\ \|\|x-x'\\| > a \to 0 \end{cases}$ $a > 0$
Hyperbolic (hk)	**Rational Quadratic (rqk)**

$K(x,y) = tanh(\propto \|x,y\| + c)$	$1 - \frac{\|\|x - x'\|\|^2}{\|\|x - x'\|\|^2 + a}$ $a > 0$
Radial Basic (rbk)	**Truncated (trk)**
$(\sum_{i=1}^{d} e = \gamma(x_i - x_i')^2)^m$ $\gamma > 0, m \in N$	$\frac{1}{d}\sum_{i}^{d} max\,(0, \frac{\|x_i - x_i'\|}{\gamma})$ $\gamma > 0$
Canberra (ck)	**Laplacian (lpl)**
$1 - \frac{1}{d}\sum_{i}^{d} \gamma \frac{\|x_i - x_i'\|}{\|x_i\| + \|x_i'\|}$ $\gamma \in (0, 1]$	$k(x,y) = e^{(\frac{-\|\|x-y\|\|}{\gamma})}$
Sigmod (sigm)	**Cosine (cos)**
$k(x,y) = \tanh(\gamma\|x,y\|)$	$k(x,y) = \frac{\|\|x-y\|\|}{\|\|x\|\|.\|\|y\|\|}$

Figure 3. Mathematical formulation of the kernel functions used in the project

The mathematical function of the Kernel is represented in figure 3 and in the following table, figure 4, its respective codification in Python language.

Encoding of Kernel functions

Kernel — **Polynomial (pk)**

```
#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
```

Kernel — **Triangular (tk)**

```
#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
```

Hyperbolic (hk)

```
#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)
```

Rational Quadratic (rqk)

```
#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)
```

Radial Basic (rbk)

```
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
```

Truncated (trk)

```
#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
```

Canberra (ck)

```
#CANBERRA
def c_canberra_kernel(gamma):
    def ck(X,Y):
        return canberra(X,Y,gamma=gamma)
    return ck
```

Laplacian (lpl)

```
#LAPLACIAN
def c_laplacian_kernel(gamma):
    def lpl(X,Y):
        return laplacian(X,Y,gamma=gamma)
    return lpl
```

Sigmod (sigm) — **Cosine (cos)**

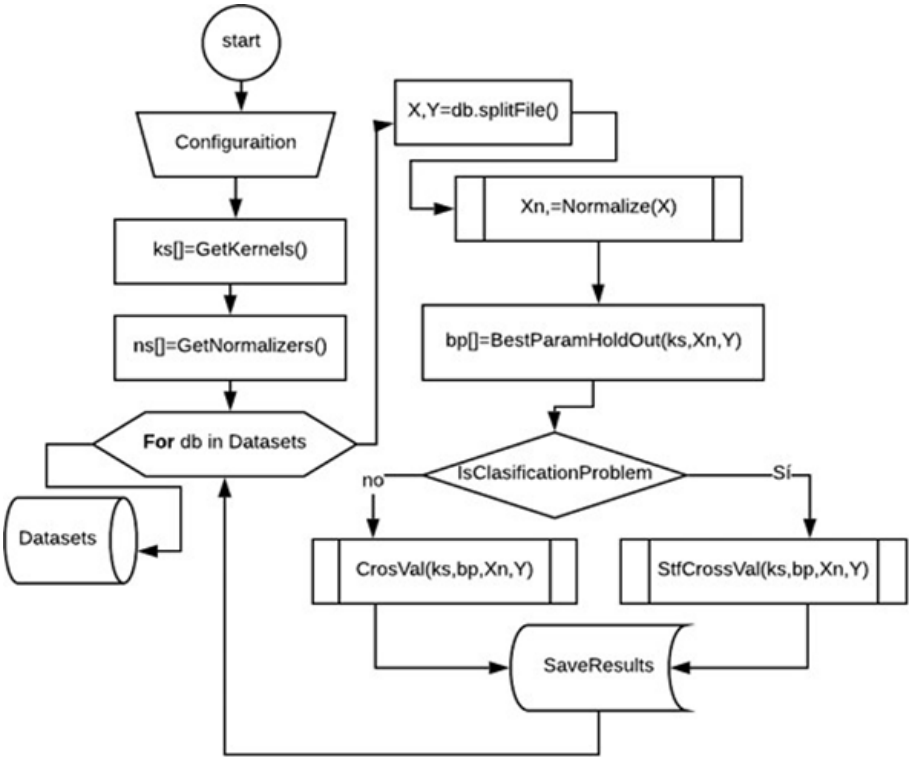
```python
# sigmoid
def sigmoid(x,y, degree=3, gamma=0.01,coef0=0.0):
    #print('x',x,'y',y)
    sm=np.tanh(gamma*x@y+coef0)
```

```python
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
```

Figure 4. Encoding of kernel functions in phyton

Therefore, the activation functions of neural networks are designed to activate and represent particular features in the data. In the cardiovascular risk prediction project, alternative kernel functions were designed to capture specific patterns in clinical or health data related to cardiovascular risk factors, such as blood pressure, weight, height, body mass index among others, once designed, these alternative Kernels would require optimization and tuning based on specific data sets in this way it could be evaluated if these Kernels improve the prediction and generalization capability compared to traditional Kernels in machine learning models.

To facilitate the codification of these kernel functions, figure 5, which illustrates the comparison algorithm using data sets, was considered.



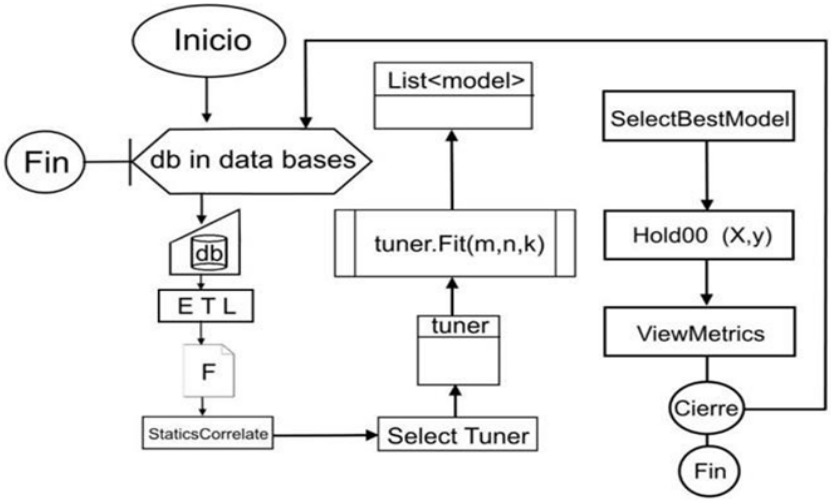
Source: https://docplayer.es/192405564-Comparativo-de-kernels-sobre-prediccion-de-oferta-de-fuentes- alternativas-de-energia.html

Figure 5. Algorithm implemented for kernel comparison

A manual configuration is performed. First, the input folder is defined, where the data to be inspected is located. The output folder is where the results of the comparison are placed. In this case, the type of problem to be processed is classification. Then, a vector of kernel functions and normalizers is obtained. We proceed to the reading of the data located in the input directory, followed by the execution of the following activities:

- A partition of the data is performed respectively.
- The normalization of the data is performed, and the parameters where the respective Kernel obtained a higher score or a lower error are identified.
- A validation is performed using these parameters. Since the problem is a classification problem, a split cross-validation is employed. Once the tests are complete, the results are recorded in the results folder.

This initial method provides a rough estimate of which Kernel function has the best performance, but to obtain a more accurate approximation, hyperparameter tuning using advanced techniques is required, so the following algorithm is established.



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

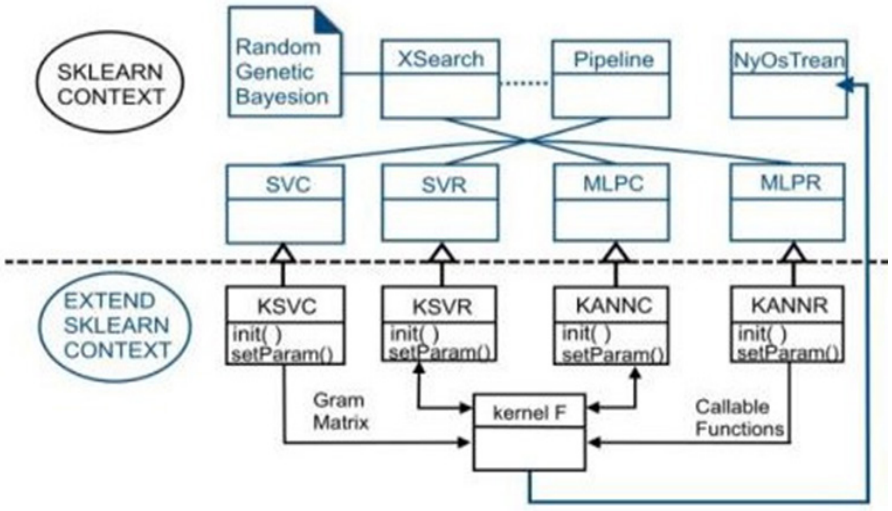
Figure 6. Exhaustive Algorithm for Model Searching

Figure 6 shows that first a cycle is executed to go through the database, extracting, transforming and loading data, which are finally synthesized in a file where the statistics and correlation aspects are found, after choosing a default tuner, configuring an evolutionary search engine (genetic algorithm), this tuner selects the best configuration of parameters, the best ones are chosen and retrained.

Kernel functions coupled to SVM and ANN

Coupling with the Scikit-learn library

To carry out and fulfill the second objective of coupling new alternative Kernel functions in SVM and ANN algorithms with the Scikit-learn library. The following was taken as a basis:



Source: https://docplayer.es/192405564-Comparativo-de-kernels-sobre-prediccion-de-oferta-de-fuentes- energy-alternatives.html

Figure 7. Coupling model Kernel functions in Scikit-learn

The Kernel trick is introduced in SVM for classification in the SVC class and further kernel functions are provided with the KSVC class via the Gram matrix as shown in equations (1) and (2):

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

$$\text{s.a}: \sum_{1}^{n} a_i y_i = 0, 0 \leq a_i \leq C, i = 1,2,\ldots,n \qquad (2)$$

A similar process is followed for the regression algorithm provided with the KSVR class whose formal expression is shown in equations (3) and (4):

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

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

For ANN, the Nystroem approximation using a basis transformation on the feature vector was used. Formally:

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

$$N_t = NyosTream(kf) \quad (6)$$

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

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

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

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

```
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
```

Figure 8. Source code of coupling to Scikit-learn library

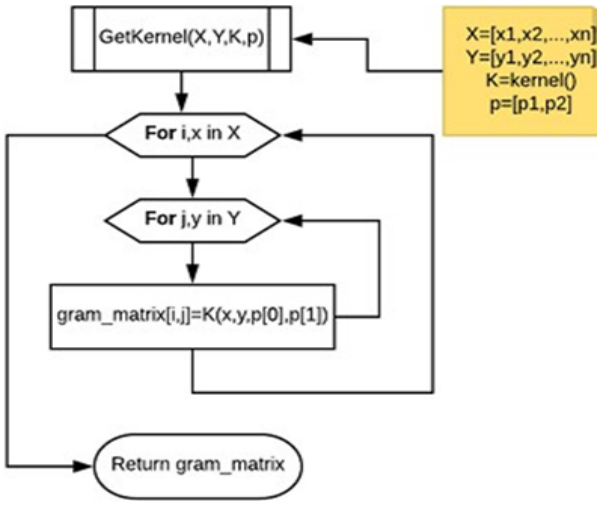
```
        if(self.kernel=="rquadratic"):
            self.kernel=rquadratic_kernel(self.degree, self.gamma, self.coef0)
            super().fit(X,y)
            return self
        if(self.kernel=="can"):
            self.kernel=canberra_kernel(self.gamma)
            super().fit(X,y)
            return 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
```

Figure 9. Source code for Scikit-learn library coupling

Obtaining Kernel functions (GetKernels procedure)

To obtain this procedure, the Gram matrix corresponding to the Kernel K function was constructed (figure 10) taking as Kernel the formulation of figure 10 and sending the corresponding parameters to each Kernel.



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

Figure 10. Algorithm for obtaining kernel functions

For the resources in figure 11, it would be enough to send as parameter to the learning algorithm (SVM, ANN).

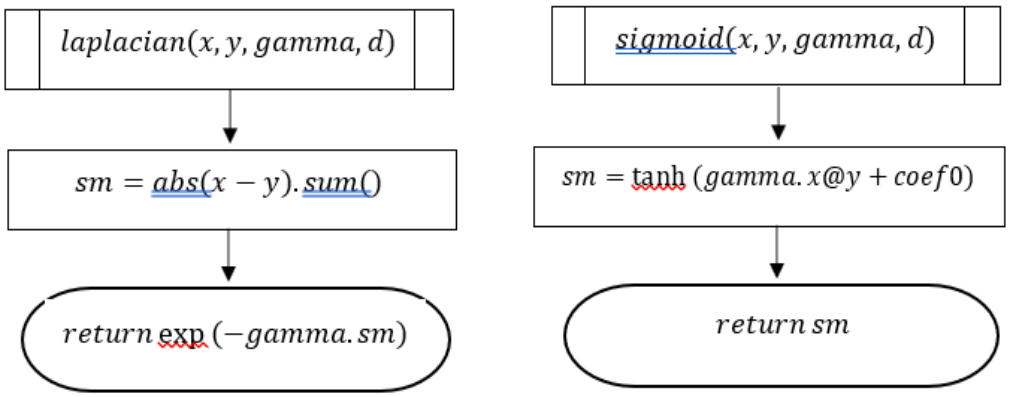


Figure 11. Example implementation of a kernel L laplacian (left), S sigmoid (right)

However, in most cases, these algorithms are designed to receive a function in their initializer (constructor) and send the training data X, Y (return functions).

Validating the proper configuration of the Kernels

In order to achieve the third objective, which is to validate the most appropriate configuration for the prevention of cardiovascular disease by comparing the relationship between error and time the results of each experiment were recorded in a dataframe, for further analysis.

Description of the data

The database used for the development of the project was a dataset consisting of 70 000 patient data records, contains 11 features plus target, and in turn consists of 3 types of entries:

- Objective: factual information.
- Examination: medical examination results.
- Subjective: information given by the patient.

Figure 12 shows the characteristics of the data.

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

Figure 12. Description of the data

Obtaining the best training parameters for SVM and ANN

In this phase, an exponential search method is used to explore the different parameter values associated with a specific Kernel within a defined range. Subsequently, the data is divided into training and test sets, with a sample size of 14,29 % reserved for the test set. Once this stage is completed, the model score is calculated using the test data, thus identifying the best parameters and their corresponding optimal score.

Kernel function inspection results in SVM classification

The following figure presents the results of evaluating the functions in classification problems to determine the adequacy of the Kernel functions. This analysis aims to validate the claim made by Mora, H., and Belanche, L., that the tasks examined in this study can lead to improved results.

Figure 13 details the results of the validation through the division into training and test sets performed with the SVM algorithm for classification, where several Kernel functions were evaluated. This process was carried out using the previously mentioned data set and applying normalization and scaling techniques.

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

Figure 13. General results for SVM

In figure 13, we have highlighted the Kernels that obtained the best score, along with the scaling comparator and the corresponding Kernel.

This scaling comparator selects the best score from figure 13 and reevaluates it using cross-validation, specifically the Kernel Truncated function. In general, excellent results were produced for this set, except for the Hyperbolic Kernel. Therefore, we will visualize the outputs of the Normalizer Scaler and Standard Scaler normalizers, as they produce better results in each Kernel.

Consequently, it can be observed that the parameter value reaches a maximum score; with this parameter, a cross-validation using different metrics is performed. Figure 14 shows the parameters used for the next stage of the algorithm.

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

Figure 14. Best parameters configured for the Kernel functions in SVM

Classification of the results

The final results of the Kernel, after configuring the best parameters described above, are summarized in figure 15, where the Kernel and the Accuracy metric are shown. This metric indicates the proportion of correct predictions made by the model compared to the total predictions, thus validating what was stated by the experts Mora.H., and Belanche.L., in which it is affirmed that the functions to be considered in the present study can obtain better results for the Support Vector Machines, thus culminating the first experimental phase.

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

Figure 15. Final results for SVM

As shown in figure 15, the Canberra Kernel achieved the highest accuracy for this dataset, with an average accuracy of 72 %.

Inspection results of Kernel functions in ANN classification

In figure 16, the results of evaluating the tasks in classification problems for ANN are presented to determine the adequacy of the Kernel functions. This analysis aims to validate the statement by experts Mora, H., and Belanche, L., who suggest that the functions to be considered worsen the results for ANN.

Presents the results of the validation, which was performed by dividing the data into training and test sets

using the ANN algorithm for classification, with several Kernel functions evaluated. This process was carried out using the previously mentioned data set and applying normalization and scaling techniques.

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
NMScaler	***triangle***	***0.711***	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

Figure 16. General results for ANN

In figure 17, we have highlighted the Kernel that obtained the best score, along with the scaling comparator and the corresponding Kernel.

This comparator selects the best score from figure 17 and reevaluates it using cross-validation, specifically the Kernel Radial Basis function in this case. In general, for this set, the results were not very good, except for the mentioned Kernel. Therefore, only the outputs of the Standard Scaler will be displayed, as it produces better results for each Kernel.

Therefore, it can be noted that the value of the parameter reaches a maximum score; with this parameter, a cross-validation with different metrics is performed. TABLE VI shows the parameters used for the input for the next stage of the algorithm.

KERNEL	BEST_PARAM	VALOR	SCORE
rquadratic	c γ Coef0	12.9154 auto 4.2813	0.684
rbf	c γ	7742.6368 0.2335	0.703
tru	c γ	7742.6368 0.2335	0.702
can	c γ	7742.6368 0.2335	0.712
radial_basic	c γ	12.9154 4.2813	***0.716***
triangle	c γ	12.9154 4.2813	0.704
hyperbolic	c γ	12.9154 4.2813	0.688
laplacian	c γ	7742.6368 0.2335	0.703
sigmoid	c γ	46.4158 545.5594	0.703
cosine	c	1.0	0.704

Figure 17. Best configured parameters for the Kernel functions in ANN

Classification of the results

The final results of the Kernel after configuration of the best parameters described above are summarized in figure 18, where the Kernel and the Accuracy metric are shown. This metric indicates the proportion of correct predictions made by the model compared to the total predictions, thus validating what was stated by the experts Mora.H., and Belanche.L., in which it is affirmed that the functions to be considered in the present study can obtain better results for the Support Vector Machines, but not for the artificial neural networks, thus culminating the first experimental phase.

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

Figure 18. Final results for ANN

As shown in the table above, the Canberra Kernel achieved the highest accuracy for this dataset, with an average accuracy of 69 %.

Once the experimental results for both SVM and ANN were obtained, the model was created with the Kernel that had the best Score. In this case, and for these data, it is more favorable to use classification algorithms based on SVM, given its performance.

Develop a web application for early warning signs of cardiovascular diseases using a predictive model.

Construction of the software tool

Since there may be users who do not have programming skills and therefore cannot use the model to evaluate the Kernels, a tool has been developed to provide an interactive and easy to use environment to perform the evaluation of new Kernels inspired by the methodology and through this, an assessment of cardiovascular risk can be made and an early warning can be obtained.

The methodology that was implemented throughout the development of the research and the objectives is the CRISP-DM methodology, it was not necessary an additional methodology to develop the software because this is perfectly part of the deployment stage, since the application is only going to consume the model and show the simple interface to the end user, for it was developed several activities presented below.

Architecture

User interaction: the user interacts with the interface, built with HTML and Bootstrap, by entering data in a form.

JavaScript validation: before submitting the form, JavaScript validates the data entered. Form submission: once the data is validated, the form is submitted to the backend using the form's action attribute.

Flask processing

- Flask receives the request and extracts the data.
- Flask sends this data to the AI model for processing.

AI model: AI model processes the data and returns the results to Flask.

Response to the frontend

- Flask takes the results from the AI model and creates an HTTP response.
- This response is sent back to the frontend.

Display results: the browser receives the HTML template with the processed results and displays them to the user.

Architectural diagram

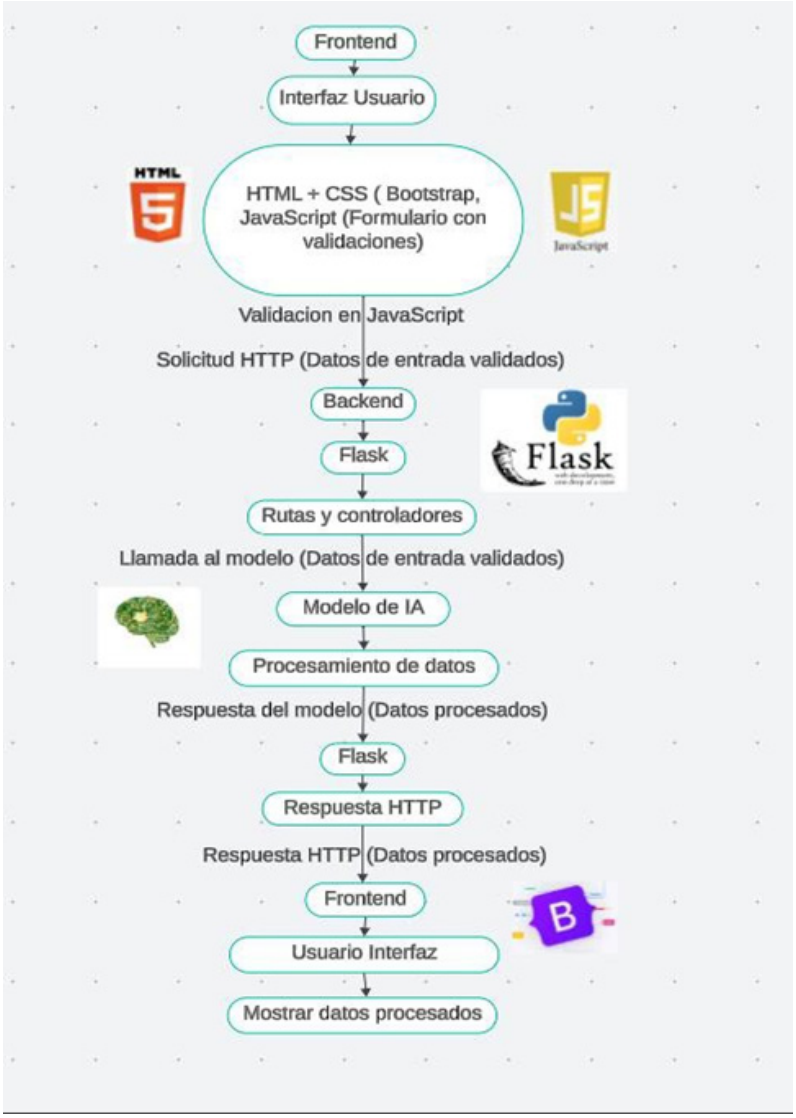


Figure 19. Architectural diagram

Model creation

At this stage, the predictive model is constructed using supervised learning techniques, specifically a Support Vector Machine (SVM) algorithm with a truncated kernel. Appropriate modeling techniques are selected for this case based on the characteristics of the data and the problem's requirements. As mentioned earlier, a Support Vector Machine (SVM) model is chosen because of its ability to handle both linear and nonlinear data sets, as well as its effectiveness in classification problems. Furthermore, by enabling probability (probability=True), the model provides probability estimates for predictions, which is helpful in risk assessment, as it is in our case.

Scenario abstraction

Within the framework of this project focused on cardiovascular risk, the health focus setting is set on the identification, prevention, and management of cardiovascular disease. This research setting applies to the entire population, such as individuals with risk factors, people with a history of cardiovascular disease, and those who could benefit from preventive interventions and who do not have cardiovascular disease. In addition, the quality of medical care, the availability of monitoring technologies, and the barriers people face in the early detection and prevention of cardiovascular events are considered. The goal is to enhance cardiovascular health by utilizing artificial intelligence models and data analysis for the early detection and implementation of preventive strategies.

Visualization of instructions

In this helpful section, descriptions are provided to facilitate a better understanding of the web application and its functions.

Analysis: functional requirements for the visualization of instructions

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

Figure 20. Functional requirement RF-01

Development

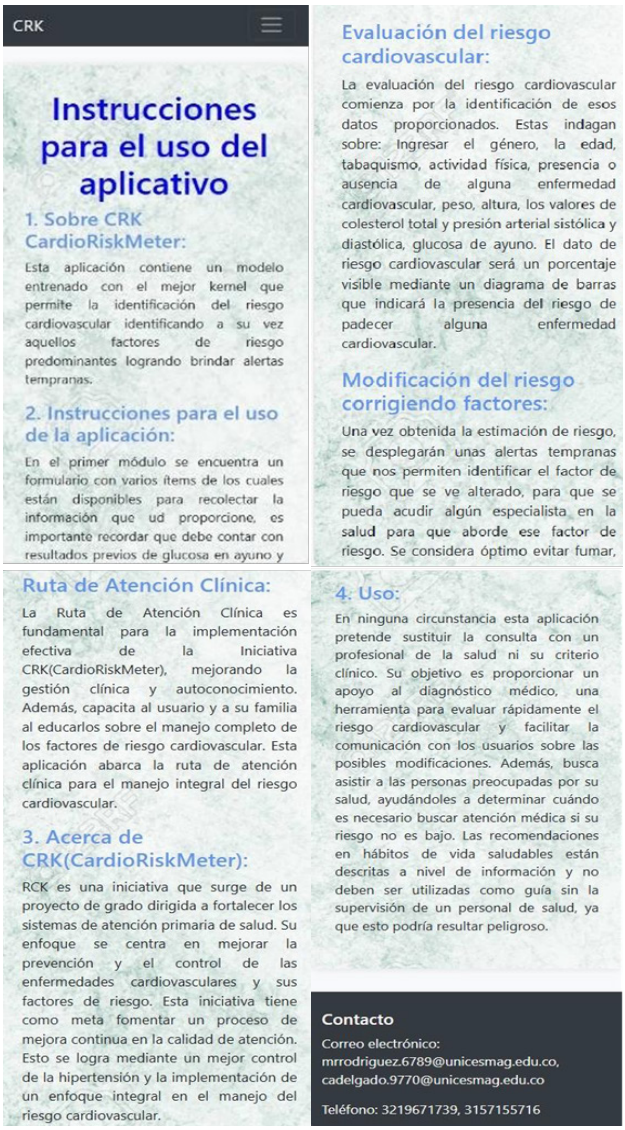


Figure 21. Instructions

The tool has clear and concise instructions to guide the user through the data entry and results visualization process. These instructions are easy to understand because it generates measurement examples and is available in a format accessible to all users.

Data selection

The web application implements a form that allows collecting the relevant data for cardiovascular disease prediction.

Analysis: functional Requirements for Cardiovascular Risk Assessment

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

Figure 22. Functional requirement RF-02

Development

The web application has an interactive form that allows the user to enter anthropometric data and cardiovascular risk factors. These data are necessary for an accurate assessment of cardiovascular disease risk.

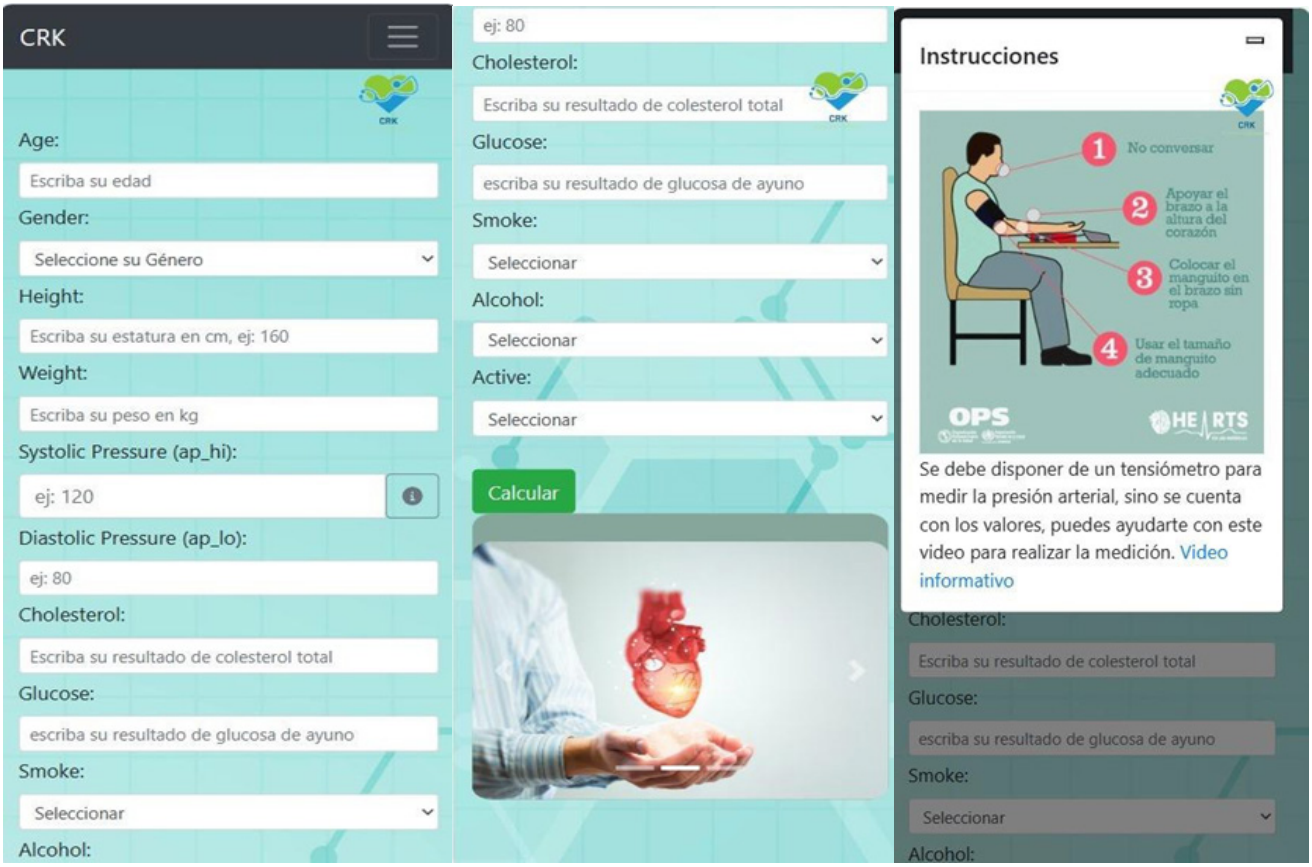


Figure 23. Data form

Cardiovascular Risk (CVR) Visualization

At this stage the user will be able to visualize the results obtained by completing the form and the percentage of suffering from a cardiovascular disease (CVD).

Analysis: functional requirements for the visualization of CVD

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

Estabilidad:	Alta
Estado:	Implementado
Comentarios:	Ninguno

Figure 24. Functional requirement RF-03

Development

In this case, the data collected from the form is evaluated, and the percentage of risk of suffering from CVD is visualized. The result can also be visualized through a graph.

For this purpose, an advanced model was developed to evaluate the risk of cardiovascular diseases, using a database of 10 000 user records. This model was trained with a binary approach, where a score of 1 indicates the presence of risk and a score of 0 indicates the absence of risk. During the training process, the model learned to identify patterns and risk factors associated with cardiovascular disease using the provided data.

The model's operation is based on generating a risk percentage. If this percentage exceeds the 50 % threshold, the model classifies the individual as being at risk for cardiovascular disease. Conversely, if the rate is less than 50 %, the model determines that the individual is not at risk. This method allows a clear and direct assessment of cardiovascular risk, facilitating informed decision-making about the user's health.

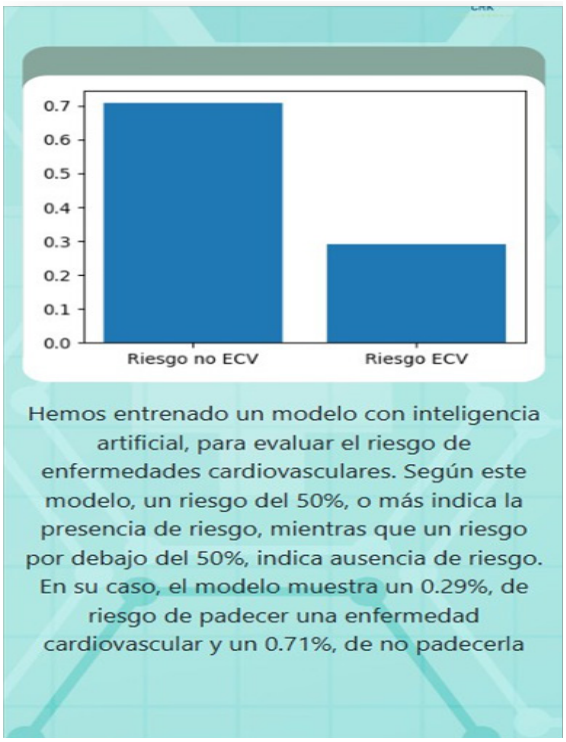


Figure 25. CVR percentage

Interpretation of risk factors generating early warnings

This section is oriented to evaluate each risk factor, allowing to generate an early warning with useful assessment information.

Analysis: Functional requirements for the visualization of early warnings

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

Figure 26. Functional requirement RF-04

Development

Early alerts are generated when there are altered form variables that allow in a certain way to generate these alerts with useful recommendations for the users.

Figure 27. Early alerts

Practical indications for a healthy lifestyle

This module contains practical advice adapted from expert recommendations for a healthy lifestyle.

Analysis: Functional requirements of practical tips

RF-05	Módulo de sugerencias prácticas
Versión:	1.0 (01-04-2023)
Autores:	Michael Rafael Rodríguez Rodríguez
	Claudia Alejandra Delgado Calpa
Descripción:	La aplicación web debe proporcionar indicaciones prácticas para la prevención de ECV
Importancia:	Esencial
Urgencia:	Inmediata
Estabilidad:	Alta
Estado:	Implementado
Comentarios:	Ninguno

Figure 28. Functional requirement RF-05

Development

The Practical Indications module aims to provide users with detailed information and practical advice on how to adopt and maintain a healthy lifestyle. This includes recommendations on nutrition, physical activity, rest, stress management, abandonment of harmful habits such as smoking and excessive alcohol consumption, as well as general guidelines for disease prevention and promotion of general well-being.

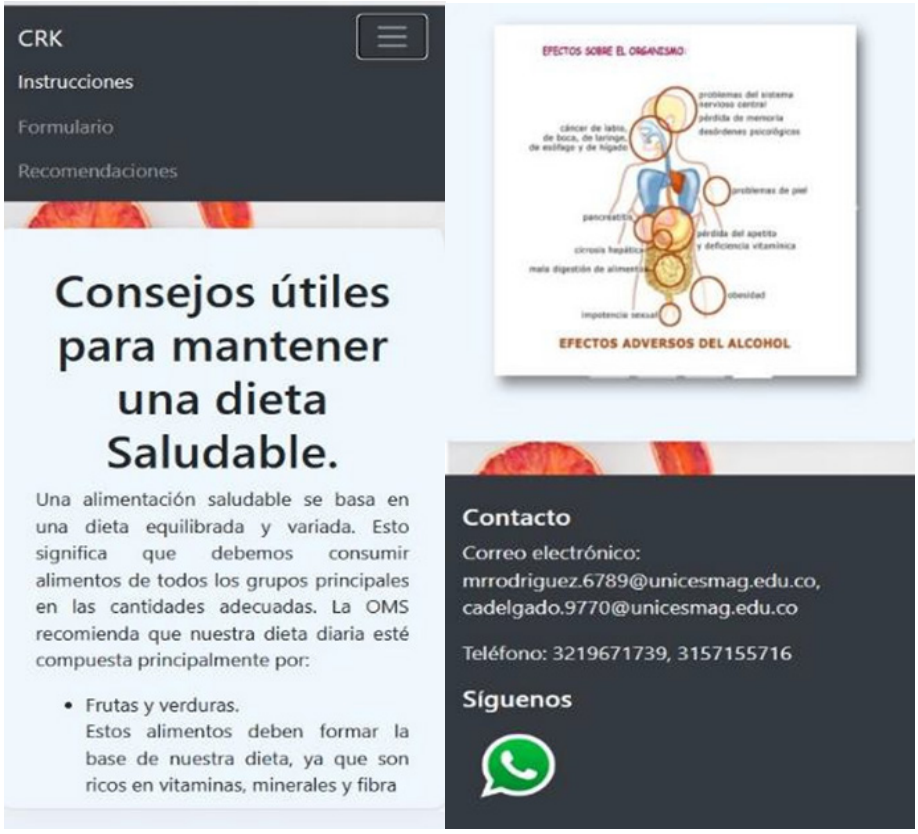


Figure 29. Recommendations

Non-functional requirements

Since the application is useful to run in different scenarios, it is important to have a good execution process with a visible graphic interface, preventing it from not adapting to the device and generating a bad user experience.

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

Figure 30. Non-functional requirement RNF-01

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

Figure 31. Non-functional requirement RNF-02

Analysis of results

In this section different trainings were carried out with various amount of data, the trainings performed were a fundamental part of the research, since it is known that there is not a kernel superior to the others, but each one has its virtues and behaviors depending on the data set to which it is being applied and its use with any machine learning algorithm, it is essential to highlight that for this case a classification algorithm was used. Following this, the model was created using the Kernel that obtained the best accuracy, and it was applied to the set of data mentioned in Table IV. Finally, the evaluation was performed with the model and the best parameters as described in TABLE VI.

In the analysis of the data obtained during the experimentation, metrics were used to measure the time required to provide results. For this purpose, Training Time was employed, which measures the total time needed to train the model from the beginning until it is completed. It includes the time required to process the data, adjust the model parameters and perform the necessary iterations, it was observed that the artificial neural networks (ANN) took less time than the support vector machines (SVM), for all Kernel, due to the Nystroem approximation, providing the results as shown in figure 32 and figure 33.

Therefore, it is essential to highlight the coupling of the models mentioned in figure 7 to the Scikit-learn library. This allowed for saving time in the construction of the hyperparameter search algorithms, thereby contributing to the software community through the development of models based on Kernel functions.

```
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
```

Figure 32. ANN time

```
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
```

Figure 33. SVM time

According to the experts, Mora, H., and Belanche, L., Kernel functions such as the Canberra and Truncated functions. Additionally, Cosine provide better results than RBF for certain types of data, confirming this hypothesis as far as this research is concerned and to verify what was stated by the experts, the results can be observed in figure 3, where the creation of these new kernels stand out for obtaining better results for SVM, than, for ANN, and independent of the configuration of the best parameters, proving that the Canberra kernel obtained 72 % of performance proving to be the most efficient option, in front of other Kernels, which can be visualized in figure 3.

From this experimentation, it can be highlighted that, depending on the configuration of the parameters, it is possible to approach and achieve better results through several comparisons, including those of more scalable Kernels, algorithms, and scaling transformers. In the case of ANNs, the comparison of scaling transformers in the Standard Scaler with the Canberra Kernel showed a performance of 71 % with the Radial Basis Kernel, and 72 % with the Canberra Kernel. For the Migmag Scaler, a percentage of 73 % was obtained with the Canberra Kernel. With the Normalizer Scaler, a rate of 71 % was obtained using the Triangular Kernel.

For the SVMs, the performance with the Standard Scaler and the Truncated Kernel was 71 %, while with the Migmag Scaler and the Truncated Kernel, it was also 71 %. With the Normalizer Scaler transformer and the Radial Basis Kernel, the performance was 72 %. A similarity in percentages was observed in these configurations, as shown in figure 3. However, in the SVMs, the Canberra Kernel proved to be the most efficient option.

Thus, it is concluded that ANNs showed a performance of less than 50 %, while SVMs exceeded 70 %. This suggests that SVM-based classification algorithms are more favorable.

One of the characteristics of the Kernels that allowed this conclusion was the study of the accuracy in Support Vector Machines where it is shown in figure 3, each Kernel with its accuracy observing that a model with accuracy values higher than 50 % is classified as acceptable which allows the development of a good model, and worsen for ANN because, each Kernel with its accuracy shows values that are below 50 % for the most part, therefore a model with less than 50 % accuracy is considered inferior meaning that the model is predicting

incorrectly more than half of the time.

Within the framework of this project, which focuses on comparing kernel functions in SVM and ANN for cardiovascular risk prediction, we recognize the need for such a model to be deployed through an application. To this end, this research environment encompasses populations such as individuals with risk factors, people with a history of cardiovascular disease, and those who could benefit from preventive interventions without suffering from cardiovascular disease. In addition, it considers the quality of medical care, the availability of monitoring technologies, and the barriers people face in the early detection and prevention of cardiovascular events. The goal is to enhance cardiovascular health by utilizing artificial intelligence models and data analysis for the early detection and implementation of preventive strategies.

The graphical tool developed in this project is based on interactive visualization of data related to cardiovascular risk. This tool was implemented using advanced data visualization technologies, such as Python, which integrates with artificial intelligence models that process and analyze clinical data from various sources. The implementation included the following stages:

1. Data were collected from different sources. These data were preprocessed to ensure quality and consistency.
2. Using Machine Learning techniques, a predictive model was developed to assess users' cardiovascular risk. These models were trained and validated using relevant data sets.
3. The graphical tool was integrated with the predictive model to enable intuitive visualization of the results. Interactive modules were developed to allow users to explore the data and gain a deeper understanding of the risk factors and model predictions.

The graphical visualization of the data enables healthcare professionals to make informed decisions based on a clear understanding of individual and collective risks, thereby supporting medical diagnosis. This is particularly useful for identifying patterns and trends that are not easily discernible through tabular data; in turn, the tool helps identify individuals at high risk of developing cardiovascular disease before obvious clinical symptoms are present. This facilitates the implementation of preventive and early intervention strategies, improving long-term health outcomes.

The graphical interface is designed to be intuitive and accessible, even for users with little experience in data analysis. This enables users and healthcare professionals to understand better the information presented and make informed, data-driven decisions. Additionally, it serves as an educational tool that clearly illustrates the effects of various risk factors and the benefits of preventive interventions.

CONCLUSIONS

The proposed experimental scenario allowed to confirm what was postulated by the experts Mora.H., and Belanche.L., and to validate the research project that the main kernel features investigated fit and behave better to the type of data provided for Support Vector Machines (SVM) and worse for Artificial Neural Networks (ANN).

The coupling of the kernel functions to the Scikit-Learn library facilitates a close relationship with developed algorithms, thereby contributing to the open-source community and academic research. This serves as a reference point for building models that integrate with this widely used library.

Visualizing the results of the experiments allowed us to evaluate the performance of each kernel objectively. This allowed us to identify the optimal parameters for the support vector machines, utilizing scaling normalizers such as the Standard Scaler and Kernel Truncated, for the classification of cardiovascular risk data, achieving a good balance between processing time and accuracy.

Finally, the development of the tool enabled the model to be graphically displayed, facilitating the learning of how to implement dynamic graphical user interfaces in Python, as well as the use of available data analysis libraries.

The developed tool is easily scalable, and its use is recommended to implement new models and functions. Additionally, it is crucial to create new visualization applications that aid in predicting cardiovascular diseases.

Since the research includes a comparison of kernel functions in both Support Vector Machines (SVM) and Artificial Neural Networks (ANN), where it was necessary to deepen in the theory of metrics used and mathematics, it is considered essential to promote among students the importance of understanding the fundamentals of these principles to address the basis of any research topic.

Finally, it is recommended to delve deeper into comparative studies of kernel functions coupled with other Machine Learning algorithms for classification or linear regression in cardiovascular disease prediction, to determine which algorithm works better with Kernels in these datasets.

BILBIOGRAPHIC REFERENCES

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.innpulsacolombia.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/saas?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, chmccl 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

FINANCING

None.

CONFLICT OF INTEREST

None.

AUTHORSHIP CONTRIBUTION

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

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

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

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

Writing - proofreading and editing: Michael Rafael Rodríguez Rodríguez, Claudia Alejandra Delgado Calpa, Héctor Andrés Mora Paz.