Implementación de Support Vector Machine para el análisis de la incidencia del dengue.

Implementation of Support Vector Machine for the analysis of dengue incidence.

José Daniel Díaz Pozo y Marcel Sebastián Lapierre Veintimilla

Resumen Este artículo se centra en la implementación del algoritmo de Support Vector Machine (SVM) optimizado con el gradiente proyectado para clasificar la incidencia del dengue en la ciudad de Guayaquil según su humedad y precipitación. Debido a los cambios climáticos ocasionados por el calentamiento global, en conjunto con el fenómeno del Niño, esta enfermedad resulta una problemática importante para la salud pública, por lo que es necesario poder clasificar de manera precisa la incidencia del dengue en la ciudad, y así, responder eficientemente a las consecuencias del virus. El estudio se basa en la recopilación de datos, realizada en colaboración con el CIP-RRD, y análisis de estos sobre la incidencia del dengue y su relación con los factores ambientales antes mencionados. Se implementaron técnicas, como el truco del kernel, para mejorar la capacidad de clasificación del algoritmo y se realizaron pruebas con el dataset de Iris y PRNN, utilizando diferentes kernels con este último. Los resultados obtenidos verificaron la robustez del algoritmo, demostrando un gran desempeño en conjuntos linealmente separables o no. Luego, se definieron los hiperparámetros adecuados para brindar una solución equilibrada, entre una clasificación precisa y un margen adecuado para el estudio de nuevos datos. Así, se constató que el algoritmo de SVM procesa de forma óptima los datos no separables linealmente a través de la utilización del kernel RBF, concluyendo que con una humedad relativa promedio mayor a 80% y una precipitación promedio mayor a 10 mm, se prevé una alta incidencia de dengue.
Palabras clave: Support Vector Machine, gradiente proyectado, dengue, optimización, aprendizaje supervisado.

Abstract This article focuses on the implementation of the Support Vector Machine (SVM) algorithm optimized with the projected gradient to classify the incidence of

José Daniel Díaz Pozo
ESPOL, Guayaquil, Ecuador, e-mail: josddiaz@espol.edu.ec.
https://orcid.org/0009-0009-2833-1171

Marcel Sebastián Lapierre Veintimilla
ESPOL, Guayaquil, Ecuador, e-mail: mslapier@espol.edu.ec.

dengue in the city of Guayaquil according to its humidity and precipitation. Due to the climate changes caused by global warming, together with the El Niño phenomenon, this disease is an important problem for public health, so it is necessary to be able to accurately classify the incidence of dengue in the city, and thus , respond efficiently to the consequences of the virus. The study is based on the collection of data, carried out in collaboration with the CIP-RRD, and analysis of these on the incidence of dengue and its relationship with the aforementioned environmental factors. Techniques, such as the kernel trick, were implemented to improve the classification capacity of the algorithm and tests were carried out with the Iris and PRNN dataset, using different kernels with the latter. The results obtained verified the robustness of the algorithm, demonstrating great performance in sets that are linearly separable or not. Appropriate hyperparameters were then defined to provide a balanced solution, between an accurate classification and an adequate margin for the study of new data. Thus, it was found that the SVM algorithm optimally processes non-linearly separable data through the use of the RBF kernel, concluding that with an average relative humidity greater than 80% and an average precipitation greater than 10 mm, it is expected a high incidence of dengue.
Keywords: Support Vector Machine, projected gradient, dengue, optimization, supervised learning.

1 Introducción

En los últimos años, el dengue ha sido una enfermedad generadora de gran preocupación debido a su propagación y sus efectos devastadores en la salud pública. La ciudad de Guayaquil no ha sido ajena a esta problemática, experimentando brotes recurrentes de dengue que han afectado a la población y han generado un impacto significativo en los sistemas de atención médica. En este contexto, la implementación de herramientas eficaces para el análisis y la predicción de la incidencia del dengue se ha convertido en una prioridad para poder enfrentar de mejor manera los brotes de esta enfermedad, convocando la participación de profesionales de la salud, investigadores, analistas, etc.

El presente estudio se enfoca en la utilización del algoritmo de Support Vector Machine (SVM) como una herramienta prometedora para el análisis de la incidencia del dengue en la ciudad de Guayaquil, pues se considera que el algoritmo SVM proporciona un separador adecuado que permita la clasificación precisa de la incidencia del virus. Para este análisis se explora específicamente el papel de las condiciones ambientales, como la humedad y la precipitación, en la propagación y el desarrollo del dengue.

El uso de SVM como enfoque principal se justifica por su capacidad para abordar problemas de clasificación complejos, donde se requiere encontrar un hiperplano óptimo que separe de manera efectiva los datos en diferentes clases. Al obtener un separador lineal óptimo, se espera mejorar la precisión en la clasificación de la incidencia del dengue, lo que a su vez permite una mejor comprensión de los factores

ambientales que influyen en la propagación de la enfermedad, desembocando en una gestión apropiada de los recursos disponibles para afrontar los casos de dengue en la ciudad.

Es por esto que el algoritmo de SVM ha sido utilizado en estudios similares, tales como los desarrollados en Tailandia, donde Siriyasatien et al. (2016) realizaron una investigación usando SVM para el pronóstico de dengue en una región del país, y Malasia, donde Salim et al. (2021), a través del análisis de variables como la temperatura, velocidad del viento, humedad y precipitación, consiguieron predecir la incidencia del dengue con una precisión del 70%. Del mismo modo, Bhatia et al. (2016) obtuvieron una precisión de 97% analizando factores ambientales con SVM en India.

También se realizó un estudio similar en China, encontrando que el algoritmo de SVM presentaba de forma consistente un menor ratio de error a la hora de predecir epidemias de dengue (Guo et al., 2017). Asimismo, Mello-Román et al. (2019) encontraron que el algoritmo SVM tenía un mejor desempeño, tanto en precisión y sensibilidad, que las redes neuronales.

Para llevar a cabo este proyecto, se recopilaron datos relevantes sobre la incidencia del dengue en Guayaquil, así como información detallada sobre las condiciones ambientales mencionadas con la ayuda del Centro Internacional del Pacífico para la Reducción de Riesgos de Desastres (CIP-RRD) cuyo objetivo es mejorar la comprensión y la gobernanza del riesgo para prevenir la aparición de nuevos riesgos de desastres y reducir los existentes, mediante la generación, uso y difusión del conocimiento, la formación profesional y la cooperación y asistencia técnica.

Con este estudio, se espera aportar conocimientos significativos a la comunidad científica y a los responsables de la toma de decisiones en salud pública. Además, se pretende sentar las bases para futuras investigaciones y el desarrollo de estrategias efectivas para la prevención y el control del dengue en Guayaquil y otras regiones afectadas. Como ya se ha hecho en Singapur, donde se han usado técnicas de machine learning para realizar alertas tempranas de brotes epidemiológicos de dengue (Shi et al., 2015).

En los siguientes capítulos se describirá el algoritmo de SVM y el gradiente proyectado, su implementación, así como las pruebas realizadas para verificar su precisión y los resultados obtenidos.

2 Descripción del problema

El presente estudio busca aplicar métodos matemáticos, específicamente el algoritmo de Support Vector Machine (SVM), para analizar y clasificar la incidencia del dengue con base en las condiciones ambientales presentes en la región, como la humedad y la precipitación.

La investigación se basa en la disponibilidad de datos relevantes sobre la incidencia del dengue en Guayaquil, así como en información detallada sobre las condiciones ambientales mencionadas. Estas se consideran de especial interés debido a su

potencial influencia en el comportamiento y la proliferación del virus del dengue, tal como menciona la Sociedad Argentina de Pediatría: ”la expansión del dengue está favorecida por el aumento de las lluvias, la humedad y el calor generados por el cambio climático.” (SAP, 2017) y Elbers et al. (2015) afirman que ”las precipitaciones y la sequía también afectan la densidad y dispersión de los mosquitos en las regiones templadas y tropicales.”. Según el Ministerio de Salud Pública, en 2023 se alcanzo la la cifra de casos de dengue más alta en los últimos siete años, llegando a ser más de 23.000 casos registrados (El Universo, 2023). A través del análisis y la clasificación de estos datos, se busca identificar posibles relaciones y patrones que permitan una mejor comprensión de la propagación del dengue en la ciudad.

No obstante, es importante destacar que esta investigación se enfrenta a restricciones y limitaciones que afectan su alcance y precisión. En primer lugar, se han encontrado dificultades significativas debido a las restricciones en los recursos económicos disponibles para la adquisición de datos adicionales y de mayor calidad. La obtención de conjuntos de datos más amplios y completos podría brindar una visión más precisa y detallada de la problemática de la clasificación de datos relacionada con el dengue. La falta de recursos financieros adecuados dificulta la posibilidad de realizar estudios exhaustivos y de obtener datos más representativos, lo que a su vez puede limitar la validez y la generalización de los resultados obtenidos.

3 Objetivos

3.1 Objetivo General

Modificar el algoritmo de Support Vector Machine aplicando el método de Gradiente proyectado para el análisis de la incidencia del virus del dengue.

3.2 Objetivos Específicos

- Recopilar datos ambientales relacionados al dengue para su posterior depurado y análisis.
- Implementar la técnica de Support Vector Machine en Python para la clasificación de la incidencia del dengue.
- Resolver el problema de optimización inherente al SVM mediante el método de gradiente proyectado.
- Determinar la regla óptima de clasificación para los datos del dengue.

4 Marco teórico

4.1 Optimización

Uno de los tópicos fundamentales del presente trabajo de investigación es la optimización, la cual "es uno de los pilares de la matemática aplicada y viene desempeñando un papel fundamental en la ingeniería". (Ayastuy, 2014)

Para un mejor entendimiento de la optimización, es importante repasar ciertos conceptos relacionados con esta, como la investigación operativa, la cual se puede definir como la implementación de métodos científicos en búsqueda de mejorar la efectividad de determinadas operaciones, decisiones y estrategias de gestión. Asimismo, la optimización puede ser categorizada como la "ciencia de aplicar los recursos disponibles para conseguir la satisfacción óptima de un objetivo específico deseado".

La optimización, consiste en la elección de la mejor alternativa entre determinadas opciones y representa una importante sección de esta ciencia, siendo esta una de las ramas que mayor evolución tuvo en el contexto de algoritmos y programación. (Ramos et al., 2010)
Los problemas de optimización están compuestos por tres elementos; la función objetivo, las variables y las restricciones, donde se busca hallar el máximo o mínimo de dicha función.

Entendiendo un problema de optimización como un sistema, podemos definir sus elementos de la siguiente manera:

- **Función objetivo:** Medida cuantitativa del funcionamiento del sistema.
- **Variables:** Medidas cambiantes que afectan la función objetivo del problema.
- **Restricciones:** Conjunto de relaciones, ya sean ecuaciones o inecuaciones, que determinan el comportamiento de las variables.

Finalmente, solucionar un problema de optimización consiste en seleccionar las variables que, siguiendo las restricciones, permitan maximizar o minimizar la función objetivo. (Ramos et al., 2010)

4.2 Optimización convexa

Dentro de la optimización podemos distinguir varios tipos de problemas con características muy marcadas, uno de ellos es el problema de optimización convexa, que será el foco de estudio de este proyecto. (Ayastuy, 2014)

La convexidad es una propiedad que puede ser extendida a funciones y conjuntos, de la siguiente manera:

Una función $f : \mathbb{R}^n \to \mathbb{R}$ es convexa si y solo si

$$\forall x, y \in \mathbb{R}^n, \forall \alpha \in [0,1] : f(\alpha x(1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y)$$

Un conjunto Q se dice que es convexo si solo si

$$\forall x, y \in Q, \forall \alpha \in [0,1] : \alpha x + (1-\alpha)y \in Q$$

Así, un problema de optimización es convexo si y solo si su función objetivo f y región factible Q son convexas. (Ayastuy, 2014)

Este tipo de problemas de optimización tienen algunas propiedades interesantes, entre las cuales resaltan las siguientes:

- Las condiciones de Karush-Kuhn-Tucker (KKT) son condiciones suficientes de optimalidad.
- El óptimo del problema dual y primal coinciden.

Primera propiedad: En esta se hace referencia a las condiciones KKT, las cuales son pruebas que determinan si un problema de optimización no lineal, puede ser resuelto obteniendo una solución óptima.

Segunda propiedad: En esta se hace referencia a las distintas formas que puede adoptar un problema de optimización convexa, siendo el primal el problema original, mientras que el dual es su contraparte estrechamente relacionada. (Ayastuy, 2014)

Es importante resaltar que uno de los problemas más estudiados dentro de la optimización convexa es el Support Vector Machine, el cual será objeto de investigación en el presente proyecto.

4.3 Support Vector Machine

El algoritmo SVM (Support Vector Machine) es uno de muchos algoritmos de ML (Machine Learning) de aprendizaje supervisado que son usados para la clasificación de datos. En este sentido, SVM ha mostrado ser un algoritmo poderoso, capaz de reconocer patrones sutiles dentro de conjuntos de datos complejos (Aruna and Rajagopalan, 2011).

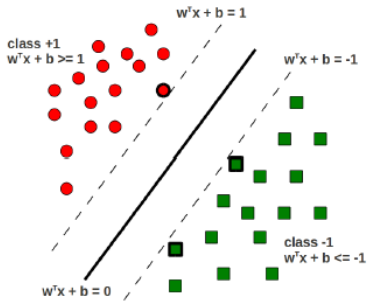
La idea central del algoritmo es conseguir un criterio de separación, también llamado hiperplano, de tal forma que se maximice la distancia entre el hiperplano y los puntos más cercanos a este, como se muestra en la figura 1. Son justamente estos puntos los que llevan el nombre de "vectores de soporte", de ahí el nombre del algoritmo.

Dado un conjunto de datos:

$$\{(x_1, y_1), \ \ldots, \ (x_n, y_n)\} \quad x_i \in X \subset \mathbb{R}^m, y_i \in \{-1, 1\}$$

donde cada y_i representa la etiqueta de x_i. El hiperplano de separación vendrá dado por $wx^T + b = 0$ y para todos los x_i se cumple la desigualdad $y_i(wx^T + b) \geq 1$ (Huanga et al., 2018).

Fig. 1: Algoritmo Support Vector Machine



Fuente: (Rai, 2018)

Como se menciona en (Vapnik et al., 1992), el margen a maximizar vendrá dado por $\frac{1}{\|w\|^2}$, lo cual implica que se debe minimizar $\|w\|^2$ o, convenientemente, $\frac{\|w\|^2}{2}$. Con lo que finalmente se obtendría el siguiente problema de optimización.

$$\begin{aligned} \min_{(x,b)} \quad & f(w,b) = \frac{\|w\|^2}{2} \\ \text{sujeto a} \quad & y_i(wx^T + b) \geq 1 \end{aligned} \tag{1}$$

4.4 Soft y Hard margin

El problema (1) se lo conoce en la literatura como ”Hard-Margin SVM” puesto que el hiperplano que se calcula busca clasificar perfectamente los datos de entrenamiento. Sin embargo, a veces es mejor otorgar cierto margen de tolerancia para que algunos valores no queden perfectamente clasificados por el hiperplano. Esto se hace porque en la aplicación, los datos no suelen ser *perfectamente separables* y es preferible evitar el *sobreajuste* del modelo (Misra, 2019).

Para añadir dicha tolerancia al problema de optimización, basta con añadir un parámetro C el cual representa el nivel de importancia que se asigna a los puntos mal clasificados, y unos valores ζ_i que serán la distancia máxima que los puntos tengan permitido estar mal clasificados (Sudheer et al., 2013)

$$\begin{aligned} \min_{(x,b)} \quad & f(w,b) = \frac{\|w\|^2}{2} + C\sum_{i=1}^{n}\xi_i \\ \text{sujeto a} \quad & y_i(wx^T + b) \geq 1 - \xi_i. \end{aligned} \tag{2}$$

Si C es un valor pequeño, el modelo intentará maximizar el margen, sin importar si esto implica que existan varios puntos mal clasificados. Por otro lado, si C es grande, el modelo evitará los errores de clasificación a costa, posiblemente, de un margen menor (Misra, 2019). Al problema (2) se lo conoce como "soft-margin SVM".

4.5 Problema Dual

Como menciona (Rai, 2018), es mas fácil y conveniente resolver el problema dual del SVM. Así, aplicando el método de Lagrange al problema (1), se obtiene el siguiente problema de optimización:

$$\begin{aligned} \max_{\alpha \geq 0} \quad & \mathcal{L}(\alpha) = \sum_{i=1}^{n} \alpha_n - \frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{n} \alpha_j \alpha_i y_j y_i (x_j^T x_i) \\ \text{sujeto a} \quad & \sum_{i=1}^{n} \alpha_i y_i = 0. \end{aligned} \tag{3}$$

Resulta fácil probar que este también se trata de un problema convexo. Lo interesante de trabajar el problema de esta forma, radica en que los puntos x_i aparecen únicamente como un producto interno. Asimismo, es posible calcular el problema dual de (2), obteniendo la siguiente expresión:

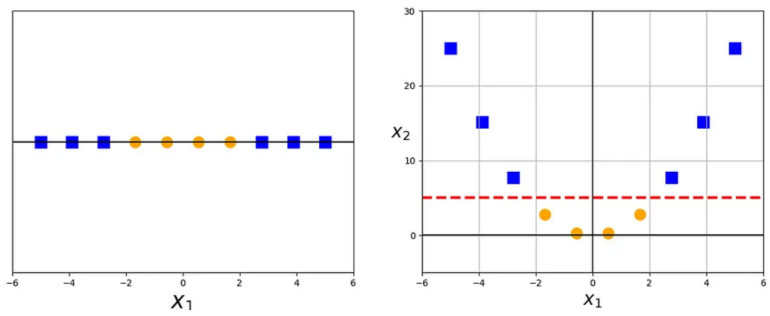
$$\begin{aligned} \max_{\alpha \leq C, \beta \geq 0} \quad & \mathcal{L}(\alpha, \beta) = \sum_{i=1}^{n} \alpha_n - \frac{1}{2} \sum_{j=1}^{n} \sum_{i=1}^{n} \alpha_j \alpha_i y_j y_i (x_j^T x_i) \\ \text{sujeto a} \quad & \sum_{i=1}^{n} \alpha_i y_i = 0. \end{aligned} \tag{4}$$

4.6 El truco del kernel

Dado que existen ocasiones donde los datos de entrenamiento no son clasificables, una solución que se suele adoptar para obtener datos separables, es recurrir a una función ϕ que mapee a cada punto del conjunto de entrenamiento a un espacio de dimensión superior, como se observa en la figura 2 (Wilimitis, 2018).

Sin embargo, la función ϕ podría llegar a ser muy compleja y extensa, dando paso al truco del kernel. Como se mencionó anteriormente, tanto la función objetivo del problema (3) como el hiperplano de clasificación dependen únicamente del producto interno, así que no es necesario calcular los valores de $\phi(x)$, únicamente basta conocer $\langle \phi(x_i), \phi(x_j) \rangle = K(x_i, x_j)$ (Wilimitis, 2018). Esta función K es llamada "kernel".

Fig. 2: Ejemplo de 2 dimensiones del truco del kernel ($\phi(x) = x^2$)



Fuente: (Wilimitis, 2018)

Existe varios tipos de kernels, en Vaerenbergh and Santamaría (2018) mencionan algunos:

- **Lineal:** $K(x_i, x_j) = x_i^T \cdot x_j$.
- **Polinomial:** $K(x_i, x_j) = (x_i^T \cdot x_j + b)^n$ con $n > 1$.
- **Funciones de Base Radial (RBF o Gaussiana):** $K(x_i, x_j) = exp(-\gamma \|x_i - x_j\|^2)$, tal que $\gamma = \frac{1}{2\sigma^2}$ donde σ representa la amplitud del kernel.
- **Funciones Sigmoide:** $K(x_i, x_j) = \tanh(\beta_0 x_i^T \cdot x_j + \beta_1)$.

4.7 Gradiente proyectado

Para poder obtener la solución óptima del problema de SVM existen distintos métodos de optimización, entre los cuales destacan aquellos que utilizan la información del gradiente (primera derivada). (Caballero et al., 2011) Por otro lado, podrían utilizarse métodos numéricos, tales como el método de Newton, sin embargo, no se consideraron puesto a que estos brindarían una aproximación, y lo que se busca es el óptimo global aprovechando las condiciones KKT.

Dentro de la optimización no lineal con restricciones los diferentes métodos pertenecen a las clases:

- Métodos de penalización exterior.
- Métodos de penalización interior.
- Métodos de proyección de gradiente.
- Método de gradiente reducido generalizado.
- Programación lineal sucesiva.
- Programación cuadrática sucesiva.

Para la resolución del problema de SVM, se escogió el método del gradiente proyectado, debido a que este posee la característica de encontrar las soluciones en un número menor de iteraciones que los métodos de las diferentes clases, esto se

debe a que ”los métodos de proyección de gradiente, cuando se aplican a problemas con restricciones lineales, proyectan el gradiente de la función objetivo en el espacio nulo de gradientes de las igualdades y desigualdades activas buscando mejorar en la dirección factible” (Caballero et al., 2011).

Sin embargo, el método del gradiente proyectado (DGP) es obtenido a partir de uno de sus semejantes, denominado el método del descenso del gradiente (DG) que, de igual manera, pertenece a la familia de métodos de optimización que utilizan la información de la primera derivada. Así, el método de gradiente proyectado, utiliza las características del método DG, aportando la posibilidad de que la solución se encuentre fuera de la región factible del problema de optimización. (Vielba, 2019)

4.8 Construcción del gradiente proyectado

El siguiente planteamiento del método del gradiente proyectado fue obtenido del libro **Linear and Nonlinear Optimization** (Luenberger and Ye, 2008)

Considere el problema de la forma:

$$\begin{aligned} min \quad & f(x) \\ \text{sujeto a} \quad & a_i^T x \leq b_i, \quad i \in I_1 \\ & a_i^T x = b_i, \quad i \in I_2 \end{aligned}$$

que tienen desigualdades e igualdades lineales.

Adicionalmente, se asumirá que el proceso de descenso se inicia en un punto factible. De esta manera, en un punto factible dado x, habrá un cierto número q de restricciones activas que satisfacen $a_i^T x = b_i$ y algunas restricciones inactivas $a_i^T x < b_i$. Inicialmente, se toma el conjunto de trabajo $W(x)$ como el conjunto de restricciones activas.

En el punto factible x, se busca un vector de dirección factible d que satisfaga $\nabla f(x)d < 0$, de modo que el movimiento en la dirección d cause una disminución en la función f.

Inicialmente, se consideran direcciones que satisfacen $a_i^T d = 0$, para $i \in W(x)$, de modo que todas las restricciones de trabajo sigan siendo activas. Este requisito equivale a requerir que el vector de dirección d se encuentre en el subespacio tangente M definido por el conjunto de restricciones de trabajo. El vector de dirección particular que se utilizará es la proyección del gradiente negativo en este subespacio.

Para calcular esta proyección, se define A_q como la matriz compuesta por las filas de las restricciones de trabajo. Suponiendo la regularidad de las restricciones, A_q será una matriz de tamaño qn con rango $q < n$. El subespacio tangente M en el cual d debe encontrarse es el subespacio de los vectores que satisfacen $A_q d = 0$. Esto significa que el subespacio N, que consiste en los vectores formados por las

filas de A_q (es decir, todos los vectores de la forma $A_q^T \lambda$ para $\lambda \in \mathbb{E}^q$), es ortogonal a M. De hecho, cualquier vector se puede escribir como la suma de vectores de cada uno de estos dos subespacios complementarios. En particular, el vector negativo del gradiente $-g_k$ puede ser escrito como

$$-g_k = d_k + A_q^T \lambda_k,$$

donde $d_k \in M$ y $\lambda_k \in E^q$. La ecuación anterior se debe resolver para λ_k considerando que $A_q d_k = 0$. Así,

$$A_q d_k = -A_q g_k - (A_q A_q^T)\lambda_k = 0$$

lo cual implica que

$$\lambda_k = -(A_q A_q^T)^{-1} A_q g_k$$

y

$$d_k = -\left[I - A_q^T (A_q A_q^T)^{-1} A_q\right] g_k = -P_k g_k$$

donde la matriz

$$P_k = \left[I - A_q^T (A_q A_q^T)^{-1} A_q\right]$$

es llamada la matriz de proyección correspondiente al subespacio M.

Se puede detectar fácilmente que si $d_k \neq 0$, entonces esta es una dirección de descenso. Dado que $g_k + d_k$ es ortogonal a d_k, se obtiene que

$$g_k^T d_k = (g_k^T + d_k^T - d_k^T) d_k = -|d_k|^2$$

Así, si d_k pasa a tomar un valor distinto de cero, esta es una dirección factible de descenso en el espacio de trabajo.

Para la elección de tamaño de paso, dado que α aumenta desde cero, el punto $x + \alpha d$ se mantendrá factible al inicio, implicando que el valor de f decrecerá. Es posible hallar la longitud del segmento factible de la línea que parte de x y luego minimizar el valor de f sobre este segmento. Si el mínimo ocurre en el punto final, una nueva restricción se activará y se añadirá al conjunto de trabajo.

Considerando la posibilidad que el negativo de la proyección de gradiente es cero, puede ser traducido como

$$\nabla f(x_k) + \lambda_k^T A_q = 0$$

donde el punto x_k satisface las condiciones necesarias de un mínimo en el espacio de trabajo. Si los componentes de λ_k correspondientes a las inecuaciones activas son todas no negativas, considerando la ecuación escrita anteriormente, es posible concluir que las condiciones KKT se satisfacen en el problema original en el punto x_k, finalizando el proceso. En este caso, el λ_k encontrado a través de la proyección del negativo del gradiente es, esencialmente, el vector del multiplicador de Lagrange para el problema original.

Si de alguna manera, al menos uno de los componentes de λ_k es negativo, es posible relajar aquella inecuación con el objetivo de moverse en una nueva dirección hacia un mejor punto. Suponiendo que λ_{jk}, la $j-$ésima componente de λ_k, es negativa

y su restricción correspondiente es la inecuación $a_j^T x \leq b$, es posible determinar la nueva dirección de vector relajando la $j-$ésima restricción y proyectando el negativo del gradiente sobre el subespacio determinado por las $q-1$ restricciones activas restantes.

Sea $A_{\bar{q}}$ la matriz A_q con una fila a_j borrada, para algún $\bar{\lambda}_k$

$$-g_k = A_q^T \lambda_k$$

$$-g_k = \bar{d}_k + A_{\bar{q}}^T \bar{\lambda}_k$$

donde $\bar{d}_k$ es la proyección de $-g_k$ usando $A_{\bar{q}}$. Es evidente que $\bar{d}_k \neq 0$, dado que las filas de A_q son linealmente independientes.

4.9 Implementación del gradiente proyectado

La siguiente implementación del método del gradiente proyectado fue obtenido del libro **Linear and Nonlinear Optimization** (Luenberger and Ye, 2008)

1. Encontrar el subespacio de restricciones activas M y forme A_q, $W(x)$.
2. Calcular $P = I - A_q^T(A_q A_q^T)^{-1} A_q$ y $d = -P\nabla f(x)^T$.
3. Si $d \neq 0$, encontrar α_1 y α_2 alcanzando, respectivamente:

$$max \ [\alpha : x + \alpha d \quad \text{es factible}]$$

$$min \ [\ f(x + \alpha d) : 0 \leq \alpha \leq \alpha_1]$$

Se debe cambiar el valor de x a $x + \alpha_2 d$ y volver al primer paso.

4. Si $d = 0$, encontrar $\lambda = -(A_q A_q^T)^{-1} A_q \nabla f(x)^T$.
   a) Si $\lambda_j \geq 0$, para todo j correspondiente a inecuaciones activas, detener el proceso; x satisface las condiciones de Karush-Kuhn-Tucker. b) En otro caso, se debe borrar la fila de A_q que corresponde a la inecuación con el componente de λ más negativo (y colocar la restricción correspondiente para $W(x)$) y volver al segundo paso.

No es necesario volver a calcular la matriz de proyección en su totalidad en cada nuevo punto. Dado que el conjunto de restricciones activas en el conjunto de trabajo cambia como máximo una restricción a la vez, es posible calcular una matriz de proyección requerida a partir de la anterior por un procedimiento de actualización.

4.9.1 Regla de Armijo

La regla de Armijo es una técnica de búsqueda lineal, esta es inexacta y se basa en el cálculo de la longitud de paso adecuado para un descenso ”suficiente” (UNA, 2019).

Dada un función diferenciable $f(x)$, x_k el punto actual, d_k la dirección de búsqueda, σ un valor fijo entre cero y uno (usualmente 0.1), ρ el factor de reducción (usualmente 0.5) y t la longitud del paso. Tendremos que la condición de Armijo es la siguiente:

$$f(x_k + td_k) \leq f(x_k) + \sigma t \nabla f(x_k) d_k$$

El proceso es sencillo, se empieza con una longitud de paso inicial t, se calcula $f(x_k + td_k)$ y si este valor es mayor que $f(x_k) + \sigma t \nabla f(x_k) d_k$, se reduce el valor de t usando el factor de reducción ($t\rho$). Se repite el proceso hasta que se cumpla la condición de Armijo.

5 Metodología

5.1 Procesamiento de los datos

Para el desarrollo del presente proyecto, se contactó con el Centro Internacional del Pacífico para la Reducción de Riesgos de Desastres (CIP-RRD) para la obtención de los datos correspondientes a la incidencia del dengue y las condiciones ambientales, estos datos corresponden a observaciones de precipitación promedio y humedad relativa del año 2017 en la ciudad de Guayaquil.

Debido a las condiciones climáticas de la zona estudiada, se consideró como un escenario altamente improbable a aquelos datos que presentaban 0 mm de precipitación promedio. Luego de este análisis, y con el objetivo de conservar únicamente la información valiosa, se descartaron los datos que provocaban inconsistencias.

Debido a la naturaleza del problema, el cual busca clasificar la incidencia del dengue según la humedad y precipitación, se concluyó que el algoritmo idóneo para su resolución es el Support Vector Machine (SVM), pues este cuenta con varias características que resultan de gran importancia en el proceso.

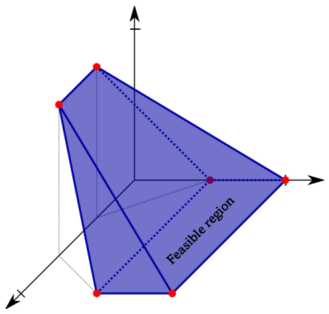
Entre ellas destaca, la capacidad y eficiencia para manejar datos no lineales. El algoritmo de SVM es conocido por su capacidad para manejar conjuntos de datos que no son linealmente separables, ya que puede transformar los datos en un espacio de mayor dimensión donde sean separables a través del truco del kernel.

5.2 Implementación del algoritmo

Una vez depurados los datos, se procedió con la implementación del código de Support Vector Machine en Python en su version 3.11 y en un sistema operativo Windows 11. Se buscó solucionar la versión dual de problema, ya que este redefine la región factible, pasando de ser un poliedro a un hiperplano, como se muestra en la figura 3.

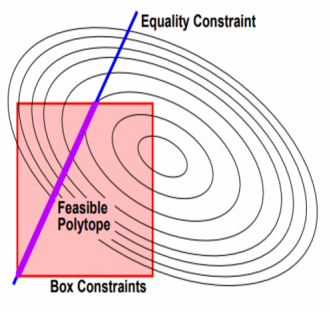
Fig. 3: Interpretación geométrica de las regiones factibles.

(a) Región factible con forma de poliedro



Adaptado de: Página web

(b) Región factible con forma de hiperplano



Fuente: (Bottou and Lin, 2007)

El algoritmo implementado fue optimizado utilizando el método del gradiente proyectado, ya que este utiliza las facilidades otorgadas por el problema dual, moviéndose por la nueva región factible sin el riesgo de salirse de la misma.

La utilización del gradiente proyectado conlleva a la resolución de un problema de maximización lineal, este problema fue resuelto a través de la librería **PuLP**, la cual brinda una cota para el tamaño del paso, evitando que la proyección deje de ser factible. Adicionalmente, el algoritmo necesita de un punto inicial factible, en este caso se seleccionó un punto inicial aleatorio que cumpla con las restricciones del problema.

Asimismo, se implementó la búsqueda lineal (***Golden search***), la cual consiste en la selección de un tamaño de paso, que minimice la función objetivo, contenido en el intervalo definido por PuLP.

La librería antes mencionada, en las ocasiones en las que el problema de maximización lineal adoptaba un comportamiento no acotado, generaba problemas al definir el tamaño de paso. Esto fue resuelto con la implementación de la **Regla de Armijo** en el código, la cual es un método de búsqueda lineal simplificado que impone cotas superiores al tamaño del paso en una dirección.

Luego, se graficaron lo puntos y se verificó que estos no eran linealmente separables, por lo que se procedió a implementar el modelo de margen suave (***soft-margin***) en el problema, con el objetivo de otorgarle más flexibilidad al modelo. Adicionalmente, se implementó el truco del kernel, con el cual se añadió el kernel polinomial y RBF al código.

5.3 Pruebas de robustez y caso de estudio

A continuación, se procedió a realizar pruebas para comprobar la robustez del algoritmo. Para ello se empleó la base de datos de *iris*, de la librería **sklearn**, y la base

de datos de *prnn_synth*, de la librería ***pmlb***, ambos procesos con un máximo de 175 iteraciones. (Romano et al., 2021)

Una vez definidos los distintos kernels, se procedió a variar el parámetro C presente en los mismos, con el objetivo de verificar cuál de ellos proporciona el menor error a la hora de clasificar. Se realizaron pruebas con el kernel lineal, polinomial y RBF, las cuales fueron graficadas a través de la librería ***matplotlib*** de Python.

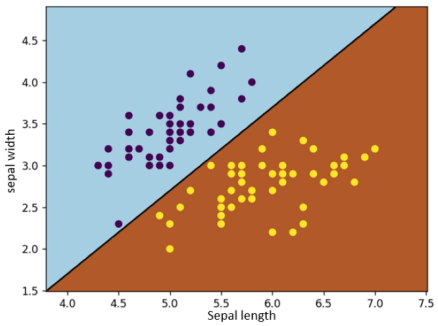
Finalmente, utilizando la técnica de **Grid Search**, se definió el kernel, el valor de C y γ que se utilizarían para abordar el problema del análisis de la incidencia del dengue en Guayaquil.

6 Resultados

6.1 Robustez del algoritmo

Con el objetivo de estudiar el desempeño del algoritmo, se realizó una prueba con el dataset de Iris, que cuenta con datos linealmente separables.

Fig. 4: Resultado del Iris dataset.

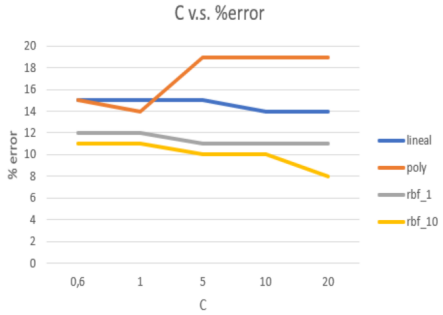


Fuente: Creación Propia

Asimismo, se realizaron pruebas con el PRNN dataset, que contiene datos no separables, variando el parámetro C en los diferentes kernels y midiendo su porcentaje de error, los resultados se muestran en la figura 5:

Adicionalmente, se muestran los tiempos de ejecución de los distintos kernels en las tablas 1 y 2.

Fig. 5: Relación entre el valor de C y el porcentaje de error.



Fuente: Creación Propia

Tabla 1: Tiempos de ejecución del código para los kernels lineal y polinomial.

Kernel	C	Tiempo (seg)
Lineal	0.6	64.58
Lineal	1	67.15
Lineal	5	148.05
Lineal	10	159.76
Lineal	20	186.59
Polinomial	0.6	201.8
Polinomial	1	187.03
Polinomial	5	204.33
Polinomial	10	204.67
Polinomial	20	204.13

Fuente: Creación Propia

6.2 Caso de estudio: Incidencia del dengue

Luego de aplicar **Grid Search** y realizar varias pruebas con distintos valores para los parámetros C y γ (gamma), se obtuvo lo siguiente:

Tabla 2: Tiempos de ejecución del código para el kernel RBF.

Kernel	C	γ	Tiempo (seg)
RBF	0.6	1	183.5
RBF	1	1	98.65
RBF	5	1	211.5
RBF	10	1	268.73
RBF	20	1	286.75
RBF	0.6	10	228.11
RBF	1	10	86.64
RBF	5	10	204.73
RBF	10	10	345.02
RBF	20	10	334.17

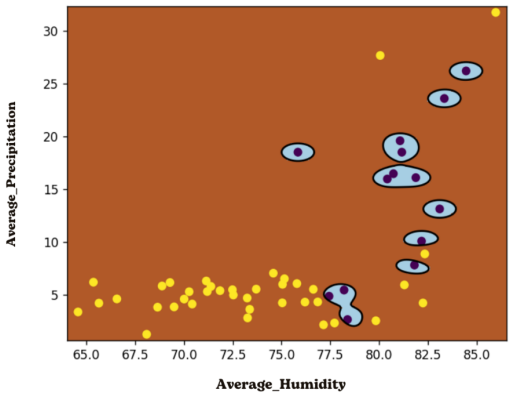
Fuente: Creación Propia

Tabla 3: Resultados representativos de la aplicación del Grid Search.

Kernel	C	γ	% error
RBF	0.6	0.25	34 %
RBF	1	0.5	23 %
RBF	7	0.25	0 %
RBF	20	10	0 %

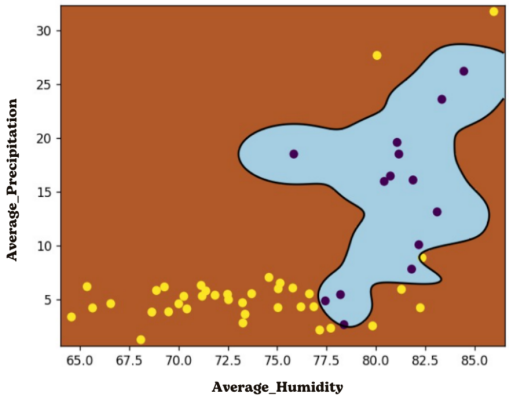
Fuente: Creación Propia

Fig. 6: Sobre ajuste causado por valores altos de los parámetros. Kernel: RBF, $C = 20$, $\gamma = 10$



Fuente: Creación Propia

Fig. 7: Resultado óptimo. Kernel: RBF, C=7, $\gamma = 0.25$



Fuente: Creación Propia

7 Análisis de resultados

Al realizar varias pruebas entorno al rendimiento del algoritmo para solucionar problemas de clasificación, es posible analizar la eficiencia del programa desde distintas perspectivas. Por un lado, cuando el conjunto de datos a estudiar es linealmente separable, el programa define satisfactoriamente el hiperplano adecuado para clasificar los puntos, tal como se observa en la figura 4.

Asimismo, en el caso de las pruebas realizadas con el dataset PRNN, el cual no es linealmente separable, el programa muestra mejores resultados con el kernel RBF. Además, es con este kernel con el cual se puede apreciar que a medida que el valor de C y de γ aumentan, el porcentaje de error disminuye, como se muestra en la figura 5. Sin embargo, un excesivo aumento de estos parámetros podría causar un sobre ajuste del modelo derivando en una falsa correcta clasificación, evidenciada en la última fila de la tabla 3, pues los puntos empezarían a ser ”encapsulados” como se muestra en la figura 6.

Adicionalmente, en la tablas 1 y 2 , se puede apreciar que un aumento en los valores de C y de γ deriva en un aumento del tiempo de ejecución del código, lo cual está estrechamente relacionado con el sobre ajuste del modelo.

Finalmente, en el caso de estudio del dengue (figura 7), se puede observar como la región turquesa representa el conjunto de puntos donde se podría presentar casos de alta incidencia del virus, misma que brinda un equilibrio entre una correcta clasificación y un margen lo suficientemente grande para incluir nuevos puntos que correspondan a una alta incidencia del dengue.

Con esta clasificación, se concluye que con una humedad relativa del 80% y una precipitación promedio de 10 mm, existirá una alta incidencia del dengue, reafirmando la existencia de una relación entre estas variables ambientales y la proliferación de mosquitos, así como con el dengue, como mencionaban (SAP, 2017).

Por otro lado, con estos resultados se pueden implementar medidas preventivas para mitigar el impacto del dengue en Guayaquil, tales como fumigaciones y campañas de concientización, así como la reestructutración de los planes para hacer frente a esta enfermedad, utilizando de mejor manera los recursos públicos.

Un ejemplo de un plan de acciones orientadas a lucha contra el dengue, podría ser el mencionado en (Martín and Brathwaite-Dick, 2007) , donde se implementó la Estrategia de Gestión Integrada para la Prevención y el Control del Dengue, también conocida como EGI-dengue, consistía en un modelo de gestión que tiene como objetivo fortalecer los programas nacionales con vistas a reducir la morbilidad, la mortalidad, la carga social y económica generada por los brotes y las epidemias de dengue, aplicada en países como Colombia, República Dominicana, Venezuela, etc.

Entre los principales ejes del modelo se encontraban:

- **Saneamiento ambiental:** contar con un grupo de trabajo ambiental multisectorial integrado y con acciones que permitan reducir los factores ambientales de riesgo de transmisión del dengue.
- **Promoción de la salud y comunicación social:** elaborar, ejecutar y evaluar planes de comunicación para lograr cambios de conducta (COMBI) en los países cubiertos por la EGI–CA-DOR.
- **Investigación y capacitación:** desarrollar investigaciones técnicas, operativas y formativas y capacitar los recursos humanos disponibles.

Con la incorporación de algoritmo del SVM a la planificación de acciones de mitigación y prevención contra el dengue, se esperaría conseguir mejores resultados que la aplicación del modelo por sí solo, el cual logró una mayor coordinación con otros sectores, especialmente con los municipios, un aumento en la capacidad de movilización de recursos y un mayor ajuste del trabajo al marco lógico convenido, el desarrollo de nuevas destrezas y habilidades en temas de participación comunitaria, educación e investigación antropológica y un aumento en la capacidad de respuesta y en la incorporación de nuevas herramientas para la vigilancia epidemiológica.

Conclusiones

- Se recopilaron, con ayuda del CIP-RRD, datos semanales de precipitación promedio y humedad relativa promedio del año 2017, asimismo, se depuraron correctamente y se eliminaron datos inconsistentes.
- Se implementó el algoritmo de Support Vector Machine en Python optimizado con el método de gradiente proyectado, mostrando un excelente desempeño tanto para conjuntos linealmente separables como para los que no lo son.
- Se implementaron técnicas para mejorar la capacidad de clasificación del modelo, tales como el truco del kernel, Golden Search, Regla de Armijo y Grid Search.
- Con una humedad relativa promedio mayor a 80% y una precipitación promedio mayor a 10 mm, se prevé una alta incidencia de dengue, con lo cual se espera

poder tomar medidas preventivas para mitigar su impacto en la salud pública de la ciudad.

Recomendaciones

- Se recomienda aumentar y mejorar la forma de recolectar los datos, con el fin de mejorar la precisión del modelo.
- Minimizar el número de operaciones que realiza el algoritmo con el objetivo de reducir sus costos computacionales y el tiempo de ejecución del código.
- Implementar un método que facilite la obtención de un mejor punto inicial factible, tal como la linealización del problema de SVM, que permita reducir el tiempo de ejecución del código.
- Automatizar la búsqueda y refinamiento de los parámetros durante la ejecución del algoritmo.
- Incorporar herramientas de Machine Learning, como la presentada de este trabajo, para la planificación de campañas de prevención y mitigación de los efectos de enfermedades tropicales como el dengue.

References

Aruna and Rajagopalan (2011). A novel SVM based CSSFFS feature selection algorithm for detecting breast cancer.

Ayastuy, A. (2014). *Optimización convexa*, pages 1–3. Universidad de Sevilla.

Bhatia, R., Jindal, A., and Sood, M. (2016). Prediction of dengue outbreak using environmental factors.

Bottou, L. and Lin, C.-J. (2007). Support Vector Machine Solvers.

Caballero, J., Ruiz-Femenia, R., and Aracil, I. (2011). *Simulación y Optimización de los Procesos Químicos*, pages 38–41. Universidad de Alicante.

El Universo (2023). Ecuador supera ya los 23.000 casos de dengue, cifra más alta en los últimos siete años. https://www.eluniverso.com/noticias/ecuador/dengue-23-mil-casos-ecuador-fenomeno-de-el-nino-ministerio-de-salud-msp-nota/.

Elbers, A., Koenraadt, C., and Meiswinkel, R. (2015). Mosquitoes and culicoides biting midges: Vector range and the influence of climate change. *Rev. Sci. Tech. Off. Int. Epizoot*, 34:123–137. http://www.agripress.be/_STUDIOEMMA_UPLOADS/downloads/1_be_0_1_2_3_4_5.pdf.

Guo, P., Liu, T., Zhang, Q., Wang, L., Xiao, J., Zhang, Q., Luo, G., Li, Z., He, J., Zhang, Y., and Ma, W. (2017). Developing a dengue forecast model using machine learning: A case study in china. *PLOS*. https://doi.org/10.1371/journal.pntd.0005973.

Huanga, S., Cai, N., Pacheco, P. P., Narandes, S., Wang, Y., and Xu, W. (2018). Applications of Support Vector Machine (SVM) Learning in Cancer Genomics.

Luenberger, D. and Ye, Y. (2008). *Linear and Nonlinear Programming*. Springer, 4 edition.

Martín, J. L. S. and Brathwaite-Dick, O. (2007). La estrategia de gestión integrada para la prevención y el control el dengue en la región de las américas. https://scielosp.org/pdf/rpsp/2007.v21n1/55-63/es.

Mello-Román, J., Mello-Román, J., Gómez-Guerrero, S., and García-Torres, M. (2019). Predictive models for the medical diagnosis of dengue: A case study in paraguay. *Hindawi*. https://doi.org/10.1155/2019/7307803.

Misra, R. (2019). Support Vector Machines-Soft Margin Formulation and Kernel Trick. https://towardsdatascience.com/support-vector-machines-soft-margin-formulation-and-kernel-trick-4c9729dc8efe.

Rai, P. (2018). Optimization (Wrap-up), and Hyperplane based Classifiers (Perceptron and Support Vector Machines).

Ramos, A., Sánchez, P., Ferrer, J. M., Barquín, J., and Linares, P. (2010). *Modelos matemáticos de optimización*, pages 3–7. Universidad Pontificia Comillas.

Romano, J. D., Le, T. T., La Cava, W., Gregg, J. T., Goldberg, D. J., Chakraborty, P., Ray, N. L., Himmelstein, D., Fu, W., and Moore, J. H. (2021). PMLB v1.0: an open source dataset collection for benchmarking machine learning methods. *arXiv preprint arXiv:2012.00058v2*.

Salim, N. A. M., Wah, Y. B., Reeves, C., Smith, M., Yaacob, W. F. W., Mudin, R. N., Dapari, R., Sapri, N. N. F. F., and Haque, U. (2021). Prediction of dengue outbreak in selangor malaysia using machine learning techniques. *Sci Rep 11*. https://doi.org/10.1038/s41598-020-79193-2.

SAP (2017). El cambio climático y las enfermedades transmitidas por vectores. https://www.sap.org.ar/uploads/archivos/general/files_cambio-climatico-mosquitos-profesionales_1574722221.pdf.

Shi, Y., Liu, X., Kok, S., Rajarethinam, J., Liang, S., Yap, G., Chong, C.-S., Lee, K.-S., Tan, S., Chin, C., Lo, A., Kong, W., Ng, L., and Cook, A. (2015). Three-month real-time dengue forecast models: An early warning system for outbreak alerts and policy decision support in singapore. *Environmental Health Perspectives*. https://doi.org/10.1289/ehp.1509981.

Siriyasatien, P., Phumee, A., Ongruk, P., Jampachaisri, K., and Kesorn, K. (2016). Analysis of significant factors for dengue fever incidence prediction. *BMC Bioinformatics*, 17:166. https://doi.org/10.1186/s12859-016-1034-5.

Sudheer, Sohani, S., Kumar, D., Malik, A., Chahar, B., Nema, A., Panigrahi, B., and Dhiman, R. (2013). A Support Vector Machine-Firefly Algorithm based forecasting model to determine malaria transmission.

UNA (2019). Introducción a la optimización con algoritmos. https://www.ing.una.py/pdf/umalca.pdf.

Vaerenbergh, V. and Santamaría (2018). Métodos kernel para clasificación. https://gtas.unican.es/files/docencia/APS/apuntes/

07_svm_kernel.pdf.

Vapnik, V., Boser, B., and Guyon, I. (1992). A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory.

Vielba, A. (2019). Métodos de optimización convexa en aprendizaje automático. pages 17–19.

Wilimitis, D. (2018). The Kernel Trick in Support Vector Classification. https://towardsdatascience.com/the-kernel-trick-c98cdbcaeb3f.