



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ciencias Naturales y Matemáticas**

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

**PROYECTO INTEGRADOR**

Previo a la obtención del Título de:

**Matemático**

Presentado por:

José Daniel Díaz Pozo

Marcel Sebastián Lapierre Veintimilla

GUAYAQUIL - ECUADOR

Año: 2023

## DEDICATORIA

A mis seres queridos que ya no están a mi lado, Papi Uno y Mami Uno, que estoy seguro les hubiera gustado ver este proyecto terminado. Y a mis hijos de cuatro patas: Pidu, Yiro y Guppy.

*José Díaz P.*

## DEDICATORIA

Dedico este trabajo a mis padres, Verónica y Benigno, fieles creyentes de mi capacidad y mis acompañantes en este gran viaje. A mi hermano, Matías, la persona que más quiero en este mundo y mi inspiración para ser siempre mejor. Por último, mi abuela y tío, Mercedes y Juan, quienes guiaron mi crecimiento y siempre me brindaron el apoyo que necesité.

*Marcel Lapierre V.*

## AGRADECIMIENTOS

A mis padres, familiares, profesores y amigos, que para bien o para mal, soy lo que soy gracias a ellos.

*José Díaz P.*

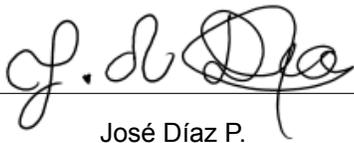
## AGRADECIMIENTOS

Quiero agradecer a mi familia, quienes siempre me acompañaron y supieron entender la complejidad del camino, sin reprochar ni juzgar. A mis mejores amigos, Daniel, Edgar, Carlos, Rivaldo, Ricardo, Marcelo, Bryan y Giancarlo, quienes han sido mi soporte en los momentos más difíciles y mis compañeros en los momentos más lindos. A Verónica, Ramón y Stefanie, que supieron ser el respaldo que necesité cuando las cosas se complicaron. Y por último, a Yogney, por su inmenso amor, apoyo y comprensión, siendo la motivación más importante en la última etapa de este viaje.

*Marcel Lapierre V.*

## DECLARACIÓN EXPRESA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; *José Daniel Díaz Pozo y Marcel Sebastián Lapierre Veintimilla* damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”



---

José Díaz P.



---

Marcel Lapierre V.

## **EVALUADORES**

---

**Luz Elimar Marchan Mendoza**

PROFESOR DE LA MATERIA

---

**Carlos Anibal Suarez Hernandez**

TUTOR

## RESUMEN

Este proyecto integrador 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, condiciones ambientales.

## **ABSTRACT**

*This project focuses on the implementation of the Support Vector Machine (SVM) algorithm optimized with the projected gradient to classify the incidence of 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, environmental conditions.*

# ÍNDICE GENERAL

RESUMEN . . . . .	I
ABSTRACT . . . . .	II
ABREVIATURAS . . . . .	V
ÍNDICE DE FIGURAS . . . . .	VI
ÍNDICE DE TABLAS . . . . .	VII
CAPÍTULO 1 . . . . .	1
1. INTRODUCCIÓN . . . . .	1
1.1 Descripción del problema . . . . .	2
1.2 Justificación del problema . . . . .	3
1.3 Objetivos . . . . .	5
1.3.1 Objetivo General . . . . .	5
1.3.2 Objetivos Específicos . . . . .	5
1.4 Marco teórico . . . . .	5
1.4.1 Optimización . . . . .	5
1.4.2 Optimización convexa . . . . .	7
1.4.3 Support Vector Machine . . . . .	8
1.4.4 Soft y Hard margin . . . . .	9
1.4.5 Problema Dual . . . . .	10
1.4.6 El truco del kernel . . . . .	11
1.4.7 Gradiente proyectado . . . . .	12

1.4.8 Construcción del gradiente proyectado . . . . .	13
1.4.9 Implementación del gradiente proyectado . . . . .	17
CAPÍTULO 2 . . . . .	18
2. METODOLOGÍA . . . . .	18
2.1 Procesamiento de los datos . . . . .	18
2.2 Implementación del algoritmo . . . . .	19
2.3 Pruebas de robustez y caso de estudio . . . . .	20
CAPÍTULO 3 . . . . .	21
3. RESULTADOS Y ANÁLISIS . . . . .	21
3.1 Resultados . . . . .	21
3.1.1 Robustez del algoritmo . . . . .	21
3.1.2 Caso de estudio: Incidencia del dengue . . . . .	23
3.2 Análisis de resultados . . . . .	25
CAPÍTULO 4 . . . . .	26
4. CONCLUSIONES Y RECOMENDACIONES . . . . .	26
BIBLIOGRAFÍA	

## **ABREVIATURAS**

ESPOL	Escuela Superior Politécnica del Litoral
SVM	Support Vector Machine
ML	Machine Learnig
KKT	Condiciones Karush-Kuhn-Tucker
RBF	Radial Basis Funtions
CIP-RRD	Centro Internacional del Pacífico para la Reducción de Riesgos de Desastres
DGP	Descent Gradient Proyected
DG	Descent Gradient

## ÍNDICE DE FIGURAS

Figura 1.1	Algoritmo Support Vector Machine . . . . .	8
Figura 1.2	Ejemplo de 2 dimensiones del truco del kernel ( $\phi(x) = x^2$ ) . . . . .	11
Figura 2.1	Interpretación geométrica de las regiones factibles. . . . .	19
Figura 3.1	Resultado del Iris dataset. . . . .	21
Figura 3.2	Relación entre el valor de $C$ y el porcentaje de error. . . . .	22
Figura 3.3	Sobre ajuste causado por valores altos de los parámetros. Kernel: RBF, $C = 20$ , $\gamma = 10$ . . . . .	24
Figura 3.4	Resultado óptimo. Kernel: RBF, $C=7$ , $\gamma = 0.25$ . . . . .	24

## ÍNDICE DE TABLAS

Tabla 3.1	Tiempos de ejecución del código para los kernels lineal y polinomial. . . . .	22
Tabla 3.2	Tiempos de ejecución del código para el kernel RBF. . . . .	23
Tabla 3.3	Resultados representativos de la aplicación del Grid Search. . . . .	23

# CAPÍTULO 1

## 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 óptimo 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.

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.

## **1.1 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. 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.

## **1.2 Justificación del problema**

La incidencia del dengue es un problema de salud pública que requiere una atención prioritaria debido a su impacto en la población y en los sistemas de salud. La ciudad de Guayaquil se enfrenta a brotes recurrentes del virus, lo que ha generado la necesidad de implementar estrategias efectivas para su prevención y control. En este sentido, el análisis de las condiciones ambientales, como la humedad y la precipitación, se ha vuelto fundamental para comprender y predecir la propagación de este virus.

Uno de los factores que se prevé influya en las condiciones ambientales de Guayaquil es el fenómeno del Niño, el cual se encuentra estrechamente relacionado con el calentamiento del océano Pacífico oriental ecuatorial y es capaz de provocar estragos en la zona intertropical y ecuatorial debido a las intensas lluvias, afectando principalmente a la región costera del Pacífico de América del Sur. Así, este fenómeno meteorológico generaría grandes cambios en el clima,

provocando alteraciones significativas en la humedad y los patrones de precipitación de la región. Estos cambios en el clima pueden tener un impacto directo en el comportamiento de la incidencia del dengue, ya que las condiciones climáticas favorables pueden propiciar la proliferación de los mosquitos portadores del virus y, por ende, el aumento de los casos de dengue.

Es crucial estar preparados para enfrentar una posible gran cantidad de casos de dengue que podrían surgir como consecuencia de las variaciones climáticas generadas por el fenómeno del Niño. Por lo tanto, resulta imperativo contar con herramientas de análisis y predicción que permitan comprender la relación entre las condiciones ambientales y la incidencia del dengue en Guayaquil. En este contexto, la aplicación del algoritmo Support Vector Machine (SVM) se presenta como una solución prometedora para la clasificación precisa de la incidencia del dengue, considerando las variables ambientales mencionadas.

Asimismo, la presente investigación se fundamenta en el interés de los investigadores por explorar y aplicar métodos matemáticos en la resolución de problemáticas sociales, culturales y de salud. Se reconoce que las matemáticas ofrecen enfoques analíticos y cuantitativos que resultan valiosos para comprender y abordar diversas situaciones problemáticas. A través del estudio de algoritmos de clasificación, como Support Vector Machine, y la aplicación del Gradiente Proyectado, se busca obtener conclusiones relevantes y contribuir al avance del conocimiento en estas áreas.

Los métodos mencionados anteriormente son de gran utilidad debido a sus características, pues el Support Vector Machine es una herramienta adecuada para la clasificación de datos separables, mientras que el gradiente proyectado cuenta con la capacidad de resolver problemas en una cantidad menor de iteraciones que los métodos de programación cuadrática.

Finalmente, esta investigación se sustenta en el propósito de establecer soluciones que fomenten la convergencia entre la computación y las ciencias exactas en Ecuador mediante la aplicación de algoritmos de Machine Learning en la clasificación de datos relacionados con el dengue. De esta manera, se aspira a sentar las bases para futuras investigaciones y proyectos que integren de manera efectiva estas disciplinas y brinden soluciones eficaces.

### **1.3 Objetivos**

#### **1.3.1 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.

#### **1.3.2 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.

### **1.4 Marco teórico**

#### **1.4.1 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)

### 1.4.2 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  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  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 y 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

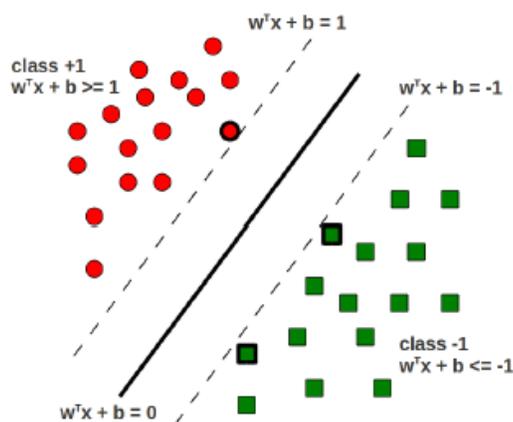
un problema de optimización convexa, siendo el primal el problema original, mientras que el dual es su contraparte estrechamente relacionada. (Ayastuy, 2014)

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.

### 1.4.3 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 1.1. Son justamente estos puntos los que llevan el nombre de “vectores de soporte”, de ahí el nombre del algoritmo.

**Figura 1.1.**  
*Algoritmo Support Vector Machine*



**Fuente:** (Rai, 2018)

Dado un conjunto de datos:

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

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).

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.

$$\begin{aligned} \min_{(w,b)} \quad & f(w, b) = \frac{\|w\|^2}{2} & (1.1) \\ \text{sujeto a} \quad & y_i(wx^T + b) \geq 1 \end{aligned}$$

#### 1.4.4 Soft y Hard margin

El problema (1.1) 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  $\zeta_i$  que serán la distancia máxima que los puntos tengan permitido estar mal clasificados (Sudheer et al., 2013)

$$\min_{(x,b)} f(w, b) = \frac{\|w\|^2}{2} + C \sum_{i=1}^n \zeta_i \quad (1.2)$$

sujeto a  $y_i(wx^T + b) \geq 1 - \zeta_i$ .

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 (1.2) se lo conoce como "soft-margin SVM".

#### 1.4.5 Problema Dual

Como menciona (Rai, 2018), es más fácil y conveniente resolver el problema dual del SVM. Así, aplicando el método de Lagrange al problema (1.1), se obtiene el siguiente problema de optimización:

$$\max_{\alpha \geq 0} \mathcal{L}(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{j=1}^n \sum_{i=1}^n \alpha_j \alpha_i y_j y_i (x_j^T x_i) \quad (1.3)$$

sujeto a  $\sum_{i=1}^n \alpha_i y_i = 0$ .

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 (1.2), obteniendo la siguiente expresión:

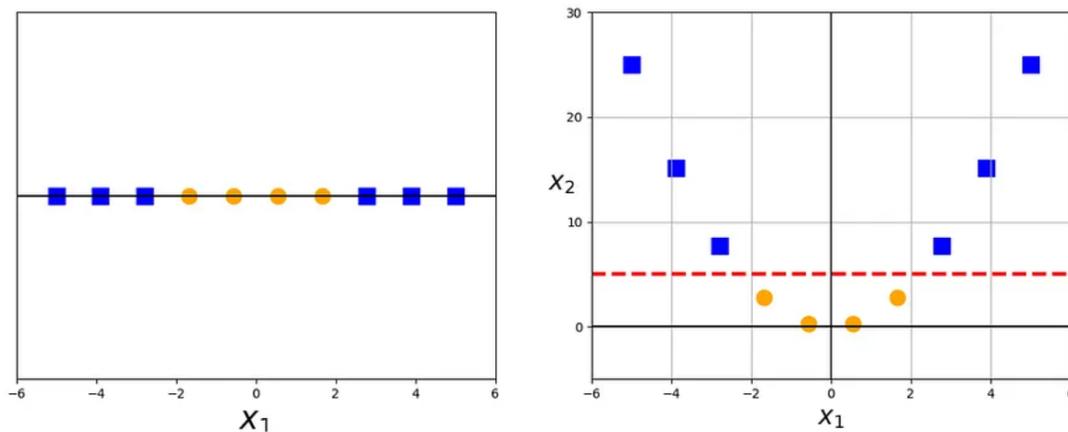
$$\begin{aligned} \max_{\alpha \leq C, \beta \geq 0} \mathcal{L}(\alpha, \beta) &= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{j=1}^n \sum_{i=1}^n \alpha_j \alpha_i y_j y_i (x_j^T x_i) \\ \text{sujeto a} \quad \sum_{i=1}^n \alpha_i y_i &= 0. \end{aligned} \quad (1.4)$$

### 1.4.6 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  $\phi$  que mapee a cada punto del conjunto de entrenamiento a un espacio de dimensión superior, como se observa en la figura 1.2 (Wilimitis, 2018).

**Figura 1.2.**

*Ejemplo de 2 dimensiones del truco del kernel ( $\phi(x) = x^2$ )*



**Fuente:** (Wilimitis, 2018)

Sin embargo, la función  $\phi$  podría llegar a ser muy compleja y extensa, dando paso al truco del kernel. Como se mencionó anteriormente, tanto la función objetivo del problema (1.3) 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". 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  $\sigma$  representa la amplitud del kernel.
- **Funciones Sigmoide:**  $K(x_i, x_j) = \tanh(\beta_0 x_i^T \cdot x_j + \beta_1)$ .

#### 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)

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)

#### **1.4.8 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  $q \times n$  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  $\lambda_k \in \mathbb{E}^q$ . La ecuación anterior se debe resolver para  $\lambda_k$  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

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

donde la matriz

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

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  $\alpha$  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  $\lambda_k$  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  $\lambda_k$  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  $\lambda_k$  es negativo, es posible relajar aquella inecuación con el objetivo de moverse en una nueva dirección hacia un mejor punto. Suponiendo que  $\lambda_{jk}$ , la  $j$ -ésima componente de  $\lambda_k$ , 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.

### 1.4.9 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  $\alpha_1$  y  $\alpha_2$  alcanzando, respectivamente:

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

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

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  $\lambda$  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.

# CAPÍTULO 2

## 2. METODOLOGÍA

### 2.1 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 en la ciudad de Guayaquil.

Una vez obtenidos los datos, se procedió a realizar una depuración a través de Microsoft Excel. Con el objetivo de conservar únicamente la información valiosa, se descartaron 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.

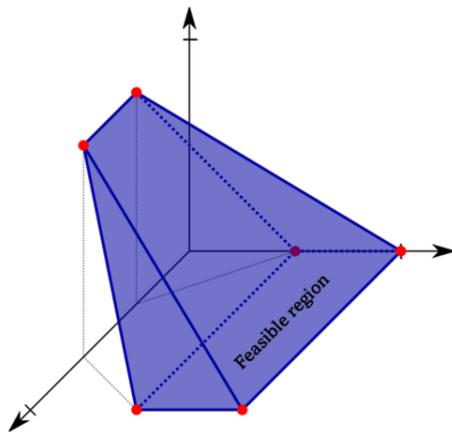
Entre ellas destaca, la capacidad y eficiencia para manejar datos no lineales. El algoritmo de SVM es conocido por su capacidad para manejar conjuntos de datos que no son linealmente separables, ya que puede transformar los datos en un espacio de mayor dimensión donde sean separables a través del truco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

## 2.2 Implementación del algoritmo

Una vez depurados los datos, se procedió con la implementación del código de Support Vector Machine en Python. 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 2.1.

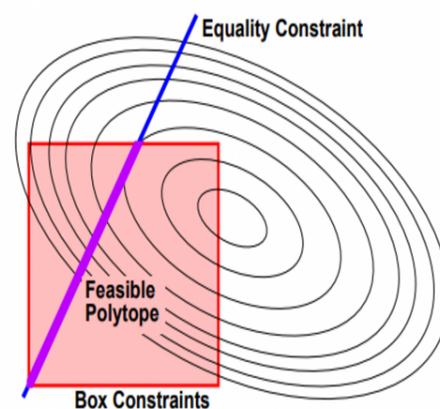
**Figura 2.1.**  
*Interpretación geométrica de las regiones factibles.*

(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.

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 los 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.

### 2.3 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  $\gamma$  que se utilizarían para abordar el problema del análisis de la incidencia del dengue en Guayaquil.

# CAPÍTULO 3

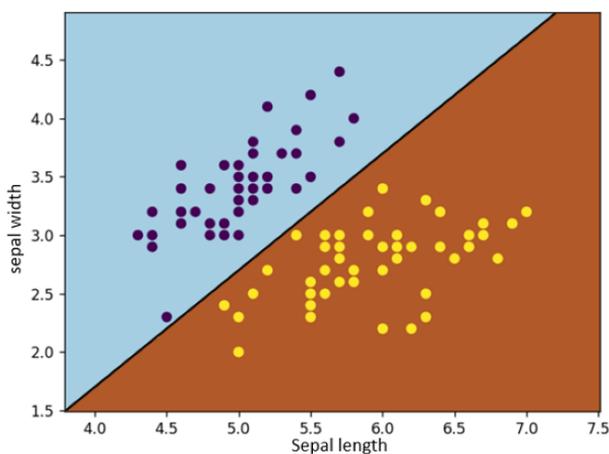
## 3. RESULTADOS Y ANÁLISIS

### 3.1 Resultados

#### 3.1.1 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.

**Figura 3.1.**  
*Resultado del Iris dataset.*

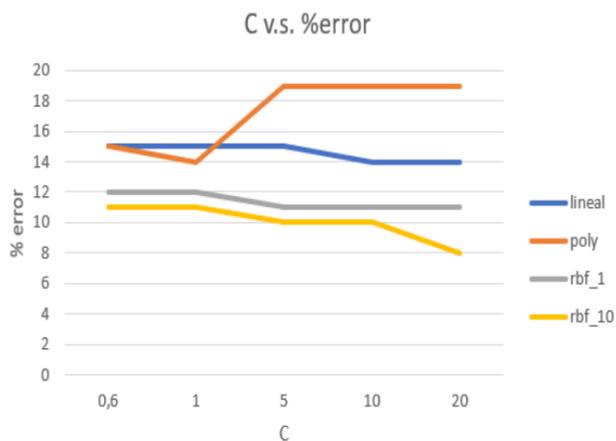


**Fuente:** Creación Propia

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, ambos resultados se muestran en la siguiente figura:

**Figura 3.2.**

Relación entre el valor de  $C$  y el porcentaje de error.



**Fuente:** Creación Propia

Adicionalmente, se muestran los tiempos de ejecución de los distintos kernels.

**Tabla 3.1.**

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

**Tabla 3.2.***Tiempos de ejecución del código para el kernel RBF.*

Kernel	C	$\gamma$	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

### 3.1.2 Caso de estudio: Incidencia del dengue

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

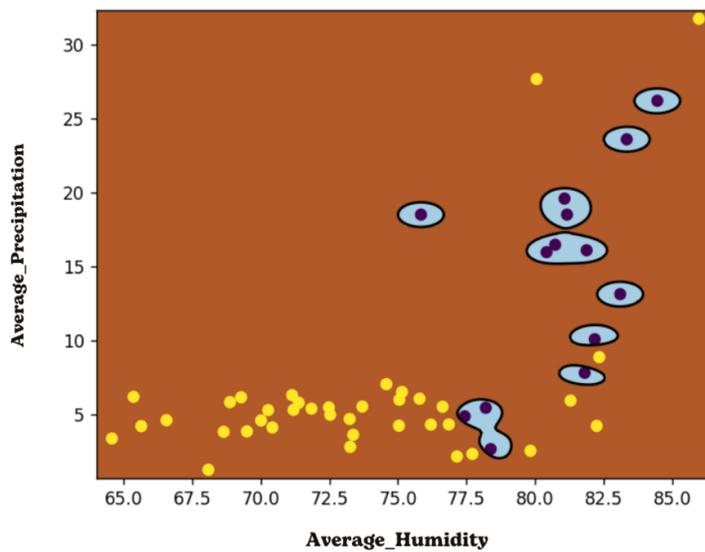
**Tabla 3.3.***Resultados representativos de la aplicación del Grid Search.*

Kernel	C	$\gamma$	% error
RBF	0.6	0.25	34 %
RBF	1	0.5	23 %
RBF	7	0.25	0 %
RBF	20	10	0 %

**Fuente:** Creación Propia

**Figura 3.3.**

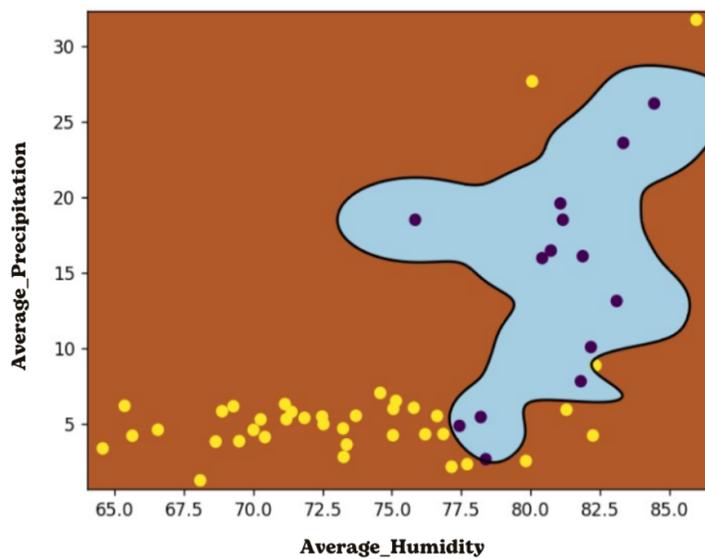
Sobre ajuste causado por valores altos de los parámetros. Kernel: RBF,  $C = 20$ ,  $\gamma = 10$



Fuente: Creación Propia

**Figura 3.4.**

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



Fuente: Creación Propia

### 3.2 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 3.1.

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  $\gamma$  aumentan, el porcentaje de error disminuye, como se muestra en la figura 3.2. 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 3.3, pues los puntos empezarían a ser "encapsulados" como se muestra en la figura 3.3.

Adicionalmente, en la tablas 3.1 y 3.2 , se puede apreciar que un aumento en los valores de  $C$  y de  $\gamma$  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 3.4), 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.

# CAPÍTULO 4

## 4. CONCLUSIONES Y RECOMENDACIONES

### Conclusiones

- Se recopilaron, con ayuda del CIP-RRD, datos semanales de temperatura y humedad 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.

## 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.

# BIBLIOGRAFÍA

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.

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.

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.

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*.
- 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.
- Vaerenbergh, V. and Santamaría (2018). Métodos kernel para clasificación. [https://gtas.unican.es/files/docencia/APS/apuntes/07\\_svm\\_kernel.pdf](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>.