**PROYECTO “SALUD CONECTADA” POR HEALTH INNOVATORS STUDENTSTEAM (HIST)**

**FASE II: PITCH 10 MEJORES DE LA REGIÓN**

Elaboró: Yessica Giraldo-Castrillon. MD. Instituto Masira-UDES

**PARTE I: ESTRUCTURACIÓN DE LA SOLUCIÓN Y PITCH**

**Introducción**

Somos **“Health Innovators StudentsTeam” (HIST),** un equipo de estudiantes de medicina del semillero del Instituto Masira, apasionados por la investigación y la innovación en salud.

Nuestro propósito es contribuir a los Objetivos de Desarrollo Sostenible (ODS) de reducir las desigualdades y mejorar la salud y el bienestar. Presentamos “Salud Conectada”, un sistema diseñado para la vigilancia y monitoreo epidemiológico de enfermedades reemergentes como el sarampión en la población migrante en Colombia. Podrá anidarse en una plataforma y funcionar por medio de una APP.

**Desarrollo**

(Fase 1: Empatizar)

Contexto y Desafíos

Personajes: María, una madre venezolana migrante, y su hijo Juan, de 3 años.

Situación: María y Juan han cruzado la frontera hacia Colombia en busca de mejores oportunidades. Sin embargo, Juan no ha recibido todas sus vacunas debido a la crisis en Venezuela.

Desafío: La falta de acceso a servicios de salud y la alta movilidad de la población migrante aumentan el riesgo de brotes de sarampión.

(Fase 2: Definir)

Problema Central

Problema: La reemergencia del sarampión en la población migrante debido a la baja cobertura de vacunación y la falta de seguimiento en el sistema de salud.

Objetivo: Desarrollar una solución que utilice datos abiertos para identificar, monitorear y vacunar a la población migrante, reduciendo así la incidencia del sarampión.

(Fase 3: Idear)

Solución Innovadora: “Salud Conectada”

Descripción: “Salud Conectada” es una plataforma digital que integra datos abiertos nacionales e internacionales para monitorear la salud de la población migrante y coordinar esfuerzos de vacunación.

Componentes Clave:

Base de Datos Integrada: Utiliza datos del Ministerio de Salud de Colombia, la Organización Panamericana de la Salud (OPS) y otras fuentes internacionales.

Aplicación Móvil: Permite a los migrantes registrar su estado de salud y recibir notificaciones sobre campañas de vacunación.

Análisis Predictivo: Utiliza algoritmos de inteligencia artificial para identificar áreas de alto riesgo y predecir posibles brotes.

Colaboración Comunitaria: Capacita a líderes comunitarios para que actúen como enlaces de salud, promoviendo la vacunación y el seguimiento médico.

(Fase 4: Prototipar)

Desarrollo del Prototipo

Interfaz de Usuario: Diseñar una aplicación móvil amigable que permita a los usuarios registrar su información de salud y recibir alertas.

Integración de Datos: Conectar la plataforma con bases de datos abiertas para obtener información en tiempo real sobre la cobertura de vacunación y los brotes de sarampión.

Capacitación: Entrenar a líderes comunitarios y personal de salud en el uso de la plataforma y la importancia de la vacunación.

(Fase 5: Evaluar)

Implementación y Evaluación

Piloto en Bucaramanga: Implementar el proyecto en 6 comunas de Bucaramanga, donde se ensamblará una cohorte de 500 adultos migrantes.

Monitoreo y Evaluación: Utilizar la plataforma para monitorear la cobertura de vacunación y la incidencia de sarampión, ajustando las estrategias según sea necesario.

Resultados Esperados:

Aumento de la Cobertura de Vacunación: Incrementar la tasa de vacunación en la población migrante.

Reducción de Brotes: Disminuir la incidencia de sarampión mediante la identificación temprana y la intervención rápida.

Fortalecimiento del Sistema de Salud: Mejorar la capacidad de respuesta ante futuras pandemias y brotes de enfermedades.

**Bases de Datos Abiertas para el Proyecto**

**Nacionales**

SIVIGILA del Ministerio de Salud de Colombia: Datos sobre cobertura de vacunación, incidencia de enfermedades, y recursos de salud.

DANE (Departamento Administrativo Nacional de Estadística): Datos demográficos y socioeconómicos.

Terridata. Creada por el DNP

Coberturas administrativas de vacunación por departamento de datos abiertos (página de MINTIC)

**Internacionales**

Epidemic Intelligence from Open Sources (EIOS): Sistema de la OMS para la detección y evaluación de amenazas de salud pública.

HealthMap: Agregador de noticias para la vigilancia de brotes en tiempo real.

ProMED: Programa para el monitoreo de enfermedades emergentes.

GeoSentinel/EuroTravNet: Red de comunicación y recopilación de datos para la vigilancia de morbilidad relacionada con viajes.

IOM (International Organization for Migration): Datos sobre movimientos migratorios.

**Metodología**

Uso de Modelos Predictivos y Machine Learning

Integración de Modelos Predictivos

Algoritmos de Machine Learning: Utilizar algoritmos como Random Forest, Support Vector Machines (SVM), y Redes Neuronales para analizar patrones en los datos y predecir brotes de enfermedades.

Análisis de Datos: Utilizar técnicas de minería de datos para identificar factores de riesgo asociados con brotes de sarampión en la población migrante.

Modelos Predictivos: Desarrollar modelos que utilicen datos climáticos, geográficos, demográficos, sociales y culturales para predecir áreas de alto riesgo.

Diseño del Algoritmo de Tamización

Variables Clave: Datos de vacunación, incidencia de enfermedades, movimientos migratorios, condiciones de hacinamiento, factores climáticos.

Proceso de Tamización: El algoritmo analizará los datos en tiempo real para identificar individuos y áreas en riesgo, generando alertas para intervenciones rápidas.

Incorporación en la Plataforma: El algoritmo se integrará en la plataforma “Salud Conectada” y la aplicación móvil, permitiendo la detección temprana y la intervención oportuna.

**Storytelling: La Historia de María y Juan**

Inicio

María y Juan llegan a Bucaramanga, enfrentando desafíos para acceder a servicios de salud.

María se entera de la aplicación “Salud Conectada” a través de un líder comunitario.

Desarrollo

María registra a Juan en la aplicación y recibe notificaciones sobre una campaña de vacunación cercana.

Juan recibe su vacuna contra el sarampión, y María recibe información sobre la importancia de la vacunación y hábitos saludables.

Clímax

Un brote de sarampión amenaza la comunidad, pero gracias a los datos de “Salud Conectada”, las autoridades de salud pueden intervenir rápidamente.

La comunidad se moviliza, y los líderes comunitarios ayudan a garantizar que todos los niños sean vacunados.

Desenlace

El brote es controlado, y la salud de Juan y otros niños está protegida.

María se convierte en una defensora de la salud en su comunidad, ayudando a otros migrantes a utilizar la plataforma.

**Impacto**

El proyecto “Salud Conectada” por Health Innovators StudentsTeam (HIST) demuestra cómo el uso innovador de datos abiertos puede transformar la salud de la población migrante en Colombia. Al integrar tecnología, colaboración comunitaria y análisis de datos, podemos crear un sistema de salud más resiliente y preparado para enfrentar futuras pandemias y brotes de enfermedades.

**PARTE II. DESARROLLO DE LA SOLUCIÓN**

**Introducción**

“Salud Conectada” es una plataforma y aplicación móvil diseñada para mejorar la vigilancia y monitoreo de enfermedades reemergentes como el sarampión en la población migrante en Colombia. Utiliza datos abiertos y algoritmos de inteligencia artificial para identificar, monitorear y vacunar a la población en riesgo.

**Objetivo:** Desarrollar un algoritmo que utilice datos abiertos y algoritmos de inteligencia artificial para identificar, monitorear y vacunar a la población migrante en riesgo de sarampión en Colombia.

**Prototipo 1.**

**Modelo Tradicional de Vigilancia Epidemiológica**

**Definiciones**

* **Caso Sospechoso:** Persona que presenta síntomas compatibles con sarampión (fiebre, erupción cutánea, tos, conjuntivitis) y tiene antecedentes de contacto con un caso confirmado o ha estado en una zona con brote de sarampión.
* **Caso Confirmado:** Persona que cumple con los criterios de un caso sospechoso y tiene confirmación de laboratorio (prueba positiva de IgM para sarampión o aislamiento del virus).

**Criterios de Valoración**

* Síntomas Clínicos: Fiebre, erupción cutánea, tos, conjuntivitis.
* Antecedentes Epidemiológicos: Contacto con casos confirmados, viaje a zonas con brote.
* Resultados de Laboratorio: Pruebas serológicas y virológicas.

**Datos Usados**

1. **Terridata del DNP:**
   * Datos Demográficos: Porcentaje de población migrante de origen venezolano en cada municipio.
   * Datos Socioeconómicos: Nivel de pobreza, acceso a servicios de salud.
2. **Sivigila:**
   * Datos Epidemiológicos: Casos reportados de sarampión.

**Métodos para Usar los Datos Combinadamente**

1. Recopilación de Datos: Se recopilan datos de Terridata y Sivigila.
2. Combinación de Datos: Los datos se combinan en un solo conjunto de datos basado en el municipio.
3. Análisis Predictivo: Utilizar algoritmos de machine learning para analizar patrones en los datos y predecir brotes de enfermedades.

**Aplicación del Algoritmo a los Datos Individuales**

1. Registro de Información: Los usuarios registran su información de salud en la aplicación móvil.
2. Análisis de Datos: El algoritmo analiza los datos individuales y los combina con los datos demográficos y epidemiológicos.
3. Generación de Alertas: El algoritmo genera alertas para individuos y áreas en riesgo, recomendando intervenciones rápidas.

**Proceso de Tamización y Detección Temprana**

1. Identificación de Riesgo: El algoritmo calcula una puntuación de riesgo basada en la población migrante y el nivel de pobreza.
2. Predicción de Casos: Utilizando modelos predictivos, el algoritmo estima el número de casos de sarampión en cada municipio.
3. Intervención Rápida: Las autoridades de salud reciben alertas y pueden intervenir rápidamente para vacunar y tratar a la población en riesgo.

**Prototipo del Modelo de Tamización y Detección Temprana**

**Datos Utilizados**

1. **Terridata del DNP**:
   * **Datos Demográficos**: Porcentaje de población migrante de origen venezolano en cada municipio.
   * **Datos Socioeconómicos**: Nivel de pobreza, acceso a servicios de salud.
2. **Sivigila**:
   * **Datos Epidemiológicos**: Casos reportados de sarampión.

**Proceso de Integración de Datos**

1. **Recopilación de Datos**: Se recopilan datos de Terridata y Sivigila.
2. **Combinación de Datos**: Los datos se combinan en un solo conjunto de datos basado en el municipio.

**Algoritmo de Tamización**

1. **Variables Clave**:
   * Población total.
   * Población migrante.
   * Nivel de pobreza.
   * Acceso a servicios de salud.
   * Casos reportados de sarampión.
2. **Diseño del Algoritmo**

**Paso 1: Recopilación y Preprocesamiento de Datos**

* **Datos Demográficos y Socioeconómicos**: Extraer datos de Terridata.
* **Datos Epidemiológicos**: Extraer datos de Sivigila.
* **Preprocesamiento**: Limpiar y normalizar los datos para su análisis.

**Paso 2: Análisis Exploratorio de Datos (EDA)**

* **Visualización**: Utilizar herramientas como R o Python para visualizar la distribución de los datos.
* **Estadísticas Descriptivas**: Calcular estadísticas descriptivas para entender mejor los datos.

**Paso 3: Desarrollo del Modelo Predictivo**

* **Selección de Algoritmos**: Considerar algoritmos como Random Forest, Gradient Boosting, o Redes Neuronales.
* **Entrenamiento del Modelo**: Dividir los datos en conjuntos de entrenamiento y prueba, y entrenar el modelo.
* **Validación del Modelo**: Evaluar el rendimiento del modelo utilizando métricas como precisión, recall y F1-score.

**Paso 4: Implementación del Algoritmo en la Aplicación Móvil**

* **Integración**: Integrar el modelo predictivo en la aplicación móvil.
* **Interfaz de Usuario**: Diseñar una interfaz intuitiva para que los usuarios registren su información de salud.
* **Generación de Alertas**: Configurar el sistema para que genere alertas basadas en los resultados del modelo.

**Paso 5: Monitoreo y Mejora Continua**

* **Monitoreo**: Supervisar el rendimiento del algoritmo y ajustar según sea necesario.
* **Actualización de Datos**: Actualizar regularmente los datos demográficos y epidemiológicos para mantener la precisión del modelo.

**8. Conclusión**

Este algoritmo permitirá una vigilancia epidemiológica más efectiva del sarampión en la población migrante en Colombia, facilitando intervenciones rápidas y mejorando la salud pública.

1. **Ejemplo de Implementación:** para este primer ejercicio se hace un caso de simulación para fines de empezar a probar el prototipado de nuestro algoritmo. Se desarrolla el script en el software libre Python.

Se usa una población hipotética de Bucaramanga y su área metropolitana.

**Modelo Predictivo**:

* **Algoritmo Utilizado**: Random Forest.
* **Entrenamiento del Modelo**: El modelo se entrena con los datos históricos para predecir el riesgo de casos de sarampión.

**Script en Python:**

import pandas as pd

import numpy as np

from sklearn.ensemble import RandomForestClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

# Datos simulados para fines de prototipo

data\_terridata = {

'municipio': ['Bucaramanga', 'Floridablanca', 'Girón', 'Piedecuesta'],

'poblacion\_total': [581130, 267849, 206932, 163622],

'poblacion\_migrante': [50000, 20000, 15000, 10000],

'nivel\_pobreza': [0.25, 0.20, 0.30, 0.22],

'acceso\_salud': [0.80, 0.85, 0.75, 0.78]

}

data\_sivigila = {

'municipio': ['Bucaramanga', 'Floridablanca', 'Girón', 'Piedecuesta'],

'casos\_sarampion': [10, 5, 8, 3],

'fecha\_reporte': ['2023-01-01', '2023-01-02', '2023-01-03', '2023-01-04']

}

# Convertir a DataFrame

df\_terridata = pd.DataFrame(data\_terridata)

df\_sivigila = pd.DataFrame(data\_sivigila)

# Combinar conjuntos de datos

df\_combined = pd.merge(df\_terridata, df\_sivigila, on='municipio')

# Crear una puntuación de riesgo basada en población migrante y nivel de pobreza

df\_combined['risk\_score'] = df\_combined['poblacion\_migrante'] \* df\_combined['nivel\_pobreza']

# Definir características y variable objetivo

X = df\_combined[['poblacion\_total', 'poblacion\_migrante', 'nivel\_pobreza', 'acceso\_salud', 'risk\_score']]

y = df\_combined['casos\_sarampion']

# Dividir datos en conjuntos de entrenamiento y prueba

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Entrenar un modelo Random Forest

model = RandomForestClassifier(n\_estimators=100, random\_state=42)

model.fit(X\_train, y\_train)

# Hacer predicciones

y\_pred = model.predict(X\_test)

# Evaluar el modelo

accuracy = accuracy\_score(y\_test, y\_pred)

print(f"Model Accuracy: {accuracy}")

# Función para predecir el riesgo de casos de sarampión en un nuevo municipio

def predict\_risk(poblacion\_total, poblacion\_migrante, nivel\_pobreza, acceso\_salud):

risk\_score = poblacion\_migrante \* nivel\_pobreza

features = np.array([[poblacion\_total, poblacion\_migrante, nivel\_pobreza, acceso\_salud, risk\_score]])

prediction = model.predict(features)

return prediction[0]

# Ejemplo de predicción para un nuevo municipio

new\_municipio\_prediction = predict\_risk(200000, 30000, 0.28, 0.80)

print(f"Predicted cases of sarampión: {new\_municipio\_prediction}")

**Resultados del Prototipo**

* **Precisión del Modelo**: La precisión del modelo es baja en este prototipo debido a la simplicidad y la cantidad limitada de datos simulados. Con datos reales y más extensos, la precisión mejorará significativamente.
* **Predicción de Casos**: El modelo puede predecir el número de casos de sarampión en un nuevo municipio basado en las características demográficas y socioeconómicas.

**Integración en la Plataforma “Salud Conectada”**

* **Aplicación Móvil**: Los usuarios pueden registrar su información de salud y recibir alertas sobre campañas de vacunación.
* **Análisis Predictivo**: El algoritmo de tamización se integrará en la plataforma para analizar los datos en tiempo real y generar alertas para intervenciones rápidas.

**Conclusión**

El prototipo del modelo de tamización y detección temprana demuestra cómo podemos utilizar datos abiertos y algoritmos de machine learning para mejorar la vigilancia epidemiológica y la respuesta a brotes de enfermedades en la población migrante. Con datos reales y más extensos, este modelo puede ser una herramienta poderosa para proteger la salud pública.

**Prototipo # 2:**

Ahora avanzaremos y diseñaremos un algoritmo moderno e innovador de vigilancia epidemiológica para el sarampión en la población migrante en Colombia, basado en inteligencia epidemiológica de campo, datos, epidemiología molecular y TIC. Luego, proporcionaré un prototipo de código que se pueda integrar en la plataforma interactiva de “Salud Conectada” y en la aplicación móvil.

**1. Diseño del Algoritmo Moderno de Vigilancia Epidemiológica**

**Componentes del Algoritmo**

1. **Recopilación de Datos en Tiempo Real**:
   * **Fuentes de Datos**: Datos demográficos y socioeconómicos (Terridata), datos epidemiológicos (Sivigila), datos de salud individual (aplicación móvil).
   * **Sensores y Dispositivos IoT**: Integración de dispositivos IoT para monitoreo de salud en tiempo real.
2. **Epidemiología Molecular**:
   * **Secuenciación Genómica**: Análisis de muestras para identificar variantes del virus del sarampión.
   * **Integración de Datos Moleculares**: Uso de datos genómicos para mejorar la precisión del modelo predictivo.
3. **Análisis Predictivo y Machine Learning**:
   * **Modelos Predictivos**: Algoritmos de machine learning (Random Forest, Gradient Boosting, Redes Neuronales) para predecir brotes.
   * **Análisis Espacio-Temporal**: Modelos para analizar la propagación del sarampión en diferentes regiones y tiempos.
4. **Generación de Alertas y Recomendaciones**:
   * **Alertas en Tiempo Real**: Notificaciones a las autoridades de salud y a los usuarios sobre posibles brotes.
   * **Recomendaciones de Intervención**: Sugerencias de acciones preventivas y de vacunación.
5. **Integración con TIC**:
   * **Plataforma Interactiva**: Integración con la plataforma “Salud Conectada” para visualización y gestión de datos.
   * **Aplicación Móvil**: Registro de información de salud y recepción de alertas por parte de los usuarios.

**2. Prototipo de Código para el Algoritmo**

A continuación, presentamos un prototipo de código en Python que ilustra cómo se podría implementar este algoritmo. Este código es un punto de partida y puede ser ampliado y adaptado según las necesidades específicas del proyecto.

**Script en Python**

import pandas as pd

import numpy as np

from sklearn.ensemble import RandomForestClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score, classification\_report

import requests

# Paso 1: Recopilación de Datos

def fetch\_data():

# Ejemplo de cómo obtener datos de Terridata y Sivigila

terridata\_url = "URL\_DE\_TERRIDATA"

sivigila\_url = "URL\_DE\_SIVIGILA"

terridata = pd.read\_csv(terridata\_url)

sivigila = pd.read\_csv(sivigila\_url)

return terridata, sivigila

# Paso 2: Combinación de Datos

def combine\_data(terridata, sivigila):

combined\_data = pd.merge(terridata, sivigila, on='municipio')

return combined\_data

# Paso 3: Preprocesamiento de Datos

def preprocess\_data(data):

# Ejemplo de preprocesamiento

data.fillna(0, inplace=True)

return data

# Paso 4: Entrenamiento del Modelo Predictivo

def train\_model(data):

X = data.drop(columns=['caso\_confirmado'])

y = data['caso\_confirmado']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = RandomForestClassifier(n\_estimators=100, random\_state=42)

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

print("Accuracy:", accuracy\_score(y\_test, y\_pred))

print(classification\_report(y\_test, y\_pred))

return model

# Paso 5: Generación de Alertas

def generate\_alerts(model, new\_data):

predictions = model.predict(new\_data)

alerts = new\_data[predictions == 1]

return alerts

# Integración con la Plataforma Interactiva y la Aplicación Móvil

def integrate\_with\_platform(alerts):

# Ejemplo de integración con la plataforma "Salud Conectada"

for index, alert in alerts.iterrows():

# Enviar alerta a la plataforma

requests.post("URL\_DE\_LA\_PLATAFORMA", data=alert.to\_dict())

# Ejecución del Algoritmo

terridata, sivigila = fetch\_data()

combined\_data = combine\_data(terridata, sivigila)

preprocessed\_data = preprocess\_data(combined\_data)

model = train\_model(preprocessed\_data)

# Simulación de nuevos datos para generar alertas

new\_data = pd.DataFrame({

'municipio': ['Municipio1', 'Municipio2'],

'porcentaje\_migrante': [30, 45],

'nivel\_pobreza': [20, 35],

'casos\_reportados': [5, 10]

})

alerts = generate\_alerts(model, new\_data)

integrate\_with\_platform(alerts)

**Prototipo # 3**

**Prototipo de Código para el Algoritmo de tamización adicionando la base de datos abiertos**

A continuación, presentamos un prototipo de código en Python que incorpora los datos de coberturas administrativas de vacunación.

**Script en Python**

import pandas as pd

import numpy as np

from sklearn.ensemble import RandomForestClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score, classification\_report

import requests

# Paso 1: Recopilación de Datos

def fetch\_data():

# Ejemplo de cómo obtener datos de Terridata y Sivigila

terridata\_url = "URL\_DE\_TERRIDATA"

sivigila\_url = "URL\_DE\_SIVIGILA"

vacunacion\_url = "https://www.datos.gov.co/resource/6i25-2hdt.csv"

terridata = pd.read\_csv(terridata\_url)

sivigila = pd.read\_csv(sivigila\_url)

vacunacion = pd.read\_csv(vacunacion\_url)

return terridata, sivigila, vacunacion

# Paso 2: Combinación de Datos

def combine\_data(terridata, sivigila, vacunacion):

combined\_data = pd.merge(terridata, sivigila, on='municipio')

combined\_data = pd.merge(combined\_data, vacunacion, on='departamento')

return combined\_data

# Paso 3: Preprocesamiento de Datos

def preprocess\_data(data):

# Ejemplo de preprocesamiento

data.fillna(0, inplace=True)

return data

# Paso 4: Entrenamiento del Modelo Predictivo

def train\_model(data):

X = data.drop(columns=['caso\_confirmado'])

y = data['caso\_confirmado']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = RandomForestClassifier(n\_estimators=100, random\_state=42)

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

print("Accuracy:", accuracy\_score(y\_test, y\_pred))

print(classification\_report(y\_test, y\_pred))

return model

# Paso 5: Generación de Alertas

def generate\_alerts(model, new\_data):

predictions = model.predict(new\_data)

alerts = new\_data[predictions == 1]

return alerts

# Integración con la Plataforma Interactiva y la Aplicación Móvil

def integrate\_with\_platform(alerts):

# Ejemplo de integración con la plataforma "Salud Conectada"

for index, alert in alerts.iterrows():

# Enviar alerta a la plataforma

requests.post("URL\_DE\_LA\_PLATAFORMA", data=alert.to\_dict())

# Ejecución del Algoritmo

terridata, sivigila, vacunacion = fetch\_data()

combined\_data = combine\_data(terridata, sivigila, vacunacion)

preprocessed\_data = preprocess\_data(combined\_data)

model = train\_model(preprocessed\_data)

# Simulación de nuevos datos para generar alertas

new\_data = pd.DataFrame({

'municipio': ['Municipio1', 'Municipio2'],

'porcentaje\_migrante': [30, 45],

'nivel\_pobreza': [20, 35],

'casos\_reportados': [5, 10],

**'cobertura\_vacunacion': [80, 90]** # Ejemplo de datos de cobertura de vacunación

})

alerts = generate\_alerts(model, new\_data)

integrate\_with\_platform(alerts)

**Conclusiones**

Este algoritmo moderno de vigilancia epidemiológica utiliza datos en tiempo real, epidemiología molecular y TIC para mejorar la detección y respuesta a brotes de sarampión en la población migrante en Colombia. El prototipo de código proporciona una base para la implementación y puede ser adaptado y ampliado según las necesidades específicas del proyecto.

Del prototipo 1 al dos, intentamos mejorar primero, el modelo teórico y el algoritmo de tamización para la vigilancia, pasando de un modelo tradicional (el actualmente usado en Colombia) de vigilancia a uno moderno, basado en la gestión desde los datos en tiempo real y usando herramientas de epidemiología de campo moderna, la cual involucra la epidemiología molecular y otras tecnologías. De esta manera, se pudo elaborar un script mejorado para mejorar a su vez el prototipo. Puede que aun haya errores, pero se observa laguna mejora en el rendimiento. El # 3 permitió mejorar cuando se incorporaron los datos abiertos disponibles. Así, podremos ir mejorando las pruebas, con las diferentes bases de datos disponibles.

Sin el mejoramiento del sistema de información, es muy difícil hacer un uso eficiente y óptimo de los datos abiertos. Por ello, **nuestra propuesta de valor es hacer una sinergia entre las potencialidades de la epidemiología moderna con la ciencia de datos y los datos abiertos, para contribuir a mejorar el sistema de información, la vigilancia epidemiológica en salud pública y favorecer la capacidad de respuesta, no solo a enfermedades reemergentes como el sarampión, sino principalmente a futuras pandemias, peores que la de la covid 19.**