FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENERÍA DE SISTEMAS E INFORMÁTICA

Tesis

Modelo de aprendizaje supervisado basado en el algoritmo XGBoost para predicción de la incidencia del dengue

Para optar el título profesional de Ingeniero de Sistemas e Informática

Autor:
Jim Harold Padilla Pierola
https://orcid.org/0000-0001-7200-8851

Asesor:
Ing. Dr. Miguel Angel Valles Coral
https://orcid.org/0000-0002-8806-2892

Coasesor:
Ing. M. Sc. Pedro Antonio Gonzales Sánchez
https://orcid.org/0000-0001-8865-7469

Tarapoto, Perú

2024

FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
ESCUELA PROFESIONAL DE INGENERÍA DE SISTEMAS E INFORMÁTICA

Tesis

Modelo de aprendizaje supervisado basado en el algoritmo XGBoost para predicción de la incidencia del dengue

Para optar el título profesional de Ingeniero de Sistemas e Informática

Autor

Jim Harold Padilla Pierola

Sustentado y aprobado el 12 de enero del 2024, por los siguientes jurados:

Presidente de Jurado
Ing. Carlos Armando Ríos López

Secretario de Jurado
Ing. John Clark San María Pinedo

Vocal de Jurado
Ing. Mg. Richard Enrique Injante Oré

Asesor
Ing. Dr. Miguel Angel Valles Coral

Coasesor
Ing. M. Sc. Pedro Antonio Gonzales Sánchez

Tarapoto, Perú

2024

Universidad Nacional de San Martín
Facultad de Ingeniería de Sistema e Informática
Jr. Vía Universitaria S/N° - Ciudad Universitaria – Morales

ACTA DE SUSTENTACIÓN
PARA OPTAR EL TÍTULO DE INGENIERO DE SISTEMAS E INFORMÁTICA

En los ambientes del Aula Magna de la Facultad de Ingeniería de Sistemas e Informática de la Universidad Nacional de San Martín, a las 18:00 horas del día viernes 12 de enero del año 2024, se reunieron los **miembros del Jurado Calificador,** integrado por:

Presidente	:	**ING. MG. CARLOS ARMANDO RÍOS LÓPEZ**
Secretario	:	**ING. JOHN CLARK. SANTA MARÍA PINEDO**
VOCAL	:	**ING. MG. RICHARD ENRIQUE INJANTE ORE**

Para evaluar la Tesis: MODELO DE APRENDIZAJE SUPERVISADO BASADO EN EL ALOGRITMO XGBOOST PARA PREDICCIÓN DE LA INCIDENCIA DEL DENGUE; presentada por el Bachiller JIM HAROLD PADILLA PIEROLA, participando en calidad de asesor el **Ing. Dr. Miguel Ángel Valles Coral,** como co asesor el Ing. M.Sc. Pedro Antonio Gonzales Sanchez.

Los señores miembros del Jurado, después de haber atendido la sustentación y evaluada las respuestas a las preguntas formuladas y terminada la réplica; luego de debatir entre sí, reservada y libremente lo declaran aprobado, por unanimidad, con el calificativo de excelente, equivalente a diecinueve (19) en fe de lo cual firmamos la presente acta, siendo las 19:00 horas del mismo día, con lo que se dio por terminado el Acto de Sustentación.

ING. MG. CARLOS ARMANDO RÍOS LÓPEZ
Presidente

ING. JOHN CLARK SANTA MARÍA PINEDO
Secretario

ING. MG. RICHARD ENRIQUE INJANTE ORE
Vocal

Declaratoria de autenticidad

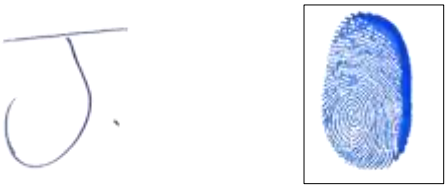
Jim Harold Padilla Pierola, con DNI N° 72167100, egresado de la Escuela Profesional de Ingeniería de Sistemas e Informática Facultad de Ingeniería de Sistemas e Informática de la Universidad Nacional de San Martín, autor de la tesis titulada: Modelo de aprendizaje supervisado basado en el algoritmo XGBoost para predicción de la incidencia del dengue.

Declaramos bajo juramento que:

1. La tesis presentada es de mi autoría.
2. La redacción fue realizada respetando las citas y referencia de las fuentes bibliográficas consultadas, siguiendo las normas APA actuales.
3. Toda información que contiene la tesis no ha sido plagiada.
4. Los datos presentados en los resultados son reales, no han sido alterados ni copiados, por tanto, la información de esta investigación debe considerarse como aporte a la realidad investigada.

Por lo antes mencionado, asumo bajo responsabilidad las consecuencias que deriven de mi accionar, sometiéndome a las leyes de nuestro país y normas vigentes de la Universidad Nacional de San Martín.

Tarapoto, 12 de enero de 2024



Jim Harold Padilla Pierola
72167100
Autor

Ficha de identificación

Título del proyecto Modelo de aprendizaje supervisado basado en el algoritmo XGBoost para predicción de la incidencia del dengue	**Área de investigación:** Ingeniería y Tecnología **Línea de investigación:** Estrategia de Tecnología de Información (TIC) **Sublínea de investigación:** Inteligencia Artificial y sistemas expertos **Grupo de investigación:** GIIA - N° 439–2022–UNSM/FISI/CFT **Tipo de investigación:** Básica ☐, Aplicada ☒, Desarrollo experimental ☐

Autor: **Jim Harold Padilla Pierola**	Facultad de Ingeniería de Sistemas e Informática Escuela Profesional de Ingeniería de Sistemas e Informática https://orcid.org/0000-0001-7200-8851
Asesor: **Ing. Dr. Miguel Angel Valles Coral**	**Dependencia local de soporte:** Facultad de Ingeniería de Sistemas e Informática Escuela Profesional de Ingeniería de Sistemas e Informática Unidad o Laboratorio Ingeniería de Sistemas e Informática https://orcid.org/0000-0002-8806-2892
Coasesor: **Ing. Dr. Sc. Pedro Antonio Gonzales Sánchez**	**Contraparte científica:** Facultad o Institución: Facultad de Ingeniería de Sistemas e Informática Unidad o Laboratorio: Ingeniería de Sistemas e Informática País: Perú https://orcid.org/0000-0001-8865-7469

Dedicatoria

Esta investigación va dedicada de manera especial para mi abuela Jacoba Celiz Villacorta, que es el principal motivo por el cuál esta investigación se desarrolló, un beso en donde quiera que estés. A mi padre y madre por siempre estar ahí para mí, confiando y apoyándome incondicionalmente, los amo.

Agradecimientos

Esta investigación es producto de mucha dedicación y esfuerzo en conjunto, a mi asesor el ing. Dr. Miguel Angel Valles Coral por ayudarme a afinar mi perspectiva de investigador y aconsejándome de la mejor manera, a mis familiares, por apoyarme en todo momento, a mis amigos cercanos, por confiar en mí, motivándome y dándome aliento para continuar, a la Universidad Nacional de San Martín, por abrirme las puertas y darme la oportunidad de haber estudiado tan bonita carrera, a mis docentes, por ayudarme a mejorar y afinar mis habilidades a lo largo de mi camino universitario. Por eso… ¡Se los agradezco!

Índice general

Índice de tablas

Índice de figuras

RESUMEN

La identificación prematura de los casos de dengue es importante para la realización de las acciones de prevención y control por parte de las entidades locales. Esta investigación, presenta una perspectiva haciendo uso de un modelo aprendizaje automático supervisado de regresión no lineal para predecir la incidencia del dengue. Se empleó un set de datos conformada por 272 periodos semanales, con datos históricos (AR) y de Google Trends (GT). El modelo fue entrenado con conjunto de datos perteneciente a 20 ciudades de Brasil con aspiración a su replicabilidad en la región San Martín, Perú. La segmentación del set de datos fue los primeros 258 periodos para entrenamiento y los 14 últimos para prueba. Se utilizó la regresión no lineal del algoritmo de “Extreme Gradient Boosting” (XGBoost), por su buen desempeño en casuísticas que no tienen proporcionalidad entre predictores y objetivo. Se emplearon técnicas para el preprocesamiento de datos, selección de características y elección de hiperparámetros para construir modelos generalizados para cada una de las 20 ciudades, en relación a solamente data histórica, Google Trends y la combinación de ambas. Los resultados obtenidos evidencian que los modelos entrenados pueden ser utilizados en 15 de las 20 ciudades. Los modelos que consumen datos de Google Trends y la combinación con datos históricos fueron los que mejores desempeños tuvieron, para la evaluación de los modelos se valió de las métricas de evaluación: RMSE, R-RMSE, R^2 y Correlación de Pearson. Evidenciando la capacidad del modelo en la predicción de la incidencia del dengue.

Palabras clave: Aprendizaje Automático, predicción, dengue, XGBoost, Google Trends.

ABSTRACT

The early identification of dengue cases is important for the implementation of prevention and control actions by local entities. This research presents a perspective using a nonlinear regression supervised machine learning model to predict the incidence of dengue. A dataset of 272 weekly periods was used, with historical data (AR) and Google Trends (GT). The model was trained with data from 20 cities in Brazil with the aim of replicating it in the San Martin region of Peru. The segmentation of the data set was the first 258 periods for training and the last 14 for testing. The non-linear regression of the "Extreme Gradient Boosting" (XGBoost) algorithm was used, due to its good performance in cases that do not have proportionality between predictors and target. Techniques for data preprocessing, feature selection and choice of hyperparameters were used to build generalized models for each of the 20 cities, in relation to historical data only, Google Trends and the combination of both. The results obtained show that the trained models can be used in 15 of the 20 cities. The models that consume data from Google Trends and the combination with historical data were the best performers. The following evaluation metrics were used to evaluate the models: RMSE, R-RMSE, R^2 and Pearson's Correlation, evidencing the capacity of the model in the prediction of dengue incidence.

Keywords: Machine Learning, prediction, dengue, XGBoost, Google Trends.

CAPÍTULO I
INTRODUCCIÓN A LA INVESTIGACIÓN

En el mundo, a consecuencia de la pandemia decretada a partir de la COVID-19, se ha tenido que replantear las estrategias para identificar la aparición de enfermedades infecciosas (Song et al., 2021) que, sin un adecuado esquema de seguimiento, ocasiona graves consecuencias en la salud de la población mundial (Pinter et al., 2020). Pero, no solo para este tipo de enfermedades, sino las infecciones endémicas que surgen debido a las deficientes políticas de salud pública que velan por el bienestar de la población (Mussumeci & Codeço Coelho, 2020; Natali et al., 2021). En el estudio de (Brett & Rohani, 2020) realizado en Puerto Rico se advierte la necesidad del desarrollo de herramientas y métodos para anticipar brotes de enfermedades infecciosas; pues ejercen una carga adicional al sistema de salud debido a la incapacidad de predecir de manera oportuna la aparición y prevalencia del dengue (da Silva Neto et al., 2022).

En Perú, se requiere conocer con presición el comportamiento de la incidencia y prevalencia del dengue (Cavany et al., 2021) para que los gobiernos regionales aborden e implementen adecuadamente las acciones de prevención y control (Elson et al., 2020). Sin embargo, ha de mencionarse que se necesitan datos disponibles, fiables y de calidad para diseñar e implementar soluciones predictivas (Ferdousi et al., 2021).

En San Martín, la data histórica recolectada de la incidencia y prevalencia del dengue no es aprovechada para la toma de decisiones de acciones preventivas para su control, debido a la imposibilidad de comprobar la veracidad de la misma (Tucto Pinedo, 2020), que además es utilizada en modelos estadísticos deterministas para la predicción de la epidemia. Así mismo, la dificultad de acceso a la información oportuna y de calidad por parte de la ciudadanía sobre la incidencia del dengue, evidencia el alejamiento de las organizaciones locales de la tecnología de la información y las comunicaciones (Lam López & Alva Arévalo, 2021).

Los responsables de epidemiología del gobierno registran datos de la incidencia, prevalencia, transmisión, tendencia y periodos de desarrollo del dengue (Aiken et al., 2020); y entienden que, con un adecuado procesamiento se podrían identificar patrones de comportamiento (Ochida et al., 2022); sin embargo, la deficiencia de este proceso impide tomar acciones preventivas e imposibilita proporcionar información a la ciudadanía; además, no se aprovechan fuentes alternativas de datos fiables para

fortalecer la estimación de brotes de la enfermedad (Amin et al., 2020) dificultando el conocimiento de la propagación de la enfermedad (Parra et al., 2020).

El empleo de modelos estadísticos deterministas que ignora el comportamiento dinámico y la heterogeneidad del problema impide predecir eficientemente la incidencia del dengue (Mudele et al., 2021), evidenciado en la dificultad para ejecutar medidas de prevención y control (Francisco et al., 2021), poniendo en tela de juicio la aplicación y calidad de estas. En el estudio de (Salami et al., 2020), los modelos de predicción contemporáneos son eficaces en el pronóstico de los brotes de la enfermedad; a pesar de esto, son impopulares en el área de epidemiología por la complejidad para desarrollar e interpretar el modelo (Ho et al., 2020), reflejado en la escasez de propuestas de modelos de este tipo.

La ausencia de una herramienta que informe la tendencia de los brotes de la enfermedad con la finalidad de divulgar oportunamente la información para la realización de acciones pertinentes por parte de las organizaciones encargadas de combatir la epidemia (McGough et al., 2021), demostró el desconocimiento de las ventajas y oportunidades que ofrece la aplicación de la tecnología como aliado estratégico para afrontar incidentes y problemas que involucran la salud (Inga-Ávila et al., 2022).

Esto se dio por el empleo de medios tradicionales para transmitir la información de acceso público, el comportamiento no determinístico de la incidencia de la enfermedad (Tsantalidou et al., 2021), y el desaprovechamiento de herramientas contemporáneas para lidiar con esta (Robles-Fernández et al., 2021), impactó en el aumento de costos en recursos humanos y materiales para la realización de las acciones preventivas y de control (Li et al., 2021), generando a mediano plazo limitaciones presupuestales y de recursos para abordar correctamente la epidemia (Xu et al., 2020).

Sucede, de acuerdo al estudio de (Salim et al., 2021), por la dificultad para realizar una propuesta empleando un modelo predictivo de alto rendimiento, y el desaprovechamiento de los datos de fuentes alternativas para reforzar la información histórica (Amin et al., 2021), requirió periodos adicionales para la detección y control de factores de riesgo del dengue (Sippy et al., 2020), reflejado en el incremento de la propagación de la enfermedad (Hoyos et al., 2021).

Esto influyó finalmente en el cumplimiento parcial de las medidas pertinentes de la incidencia y prevalencia del dengue, de continuar esta circunstancia (Sun et al., 2022), es inminente el aumento del impacto de la enfermedad sobre el sistema de salud local, esto se da por las dificultad para aprovechar eficientemente la información (Liu et al.,

2022), las complicaciones para identificar patrones de comportamiento de la epidemia (Zhao et al., 2020) y el desconocimiento de la importancia de los datos de fuentes alternativas para fortalecer los existentes (Souza et al., 2022).

Después de verificar las instituciones del estado en la región San Martín encargadas de recopilar y validar la información de la incidencia del dengue, se identificó una limitante. Era necesario acceder a datos históricos y emplear una fuente alternativa para reforzar la información recopilada, esto es de vital importancia para procurar realizar predicciones fiables, por esta razón, se decidió emplear un conjunto de datos perteneciente a la investigación realizada por (Koplewitz et al., 2022), elaborada con fuentes de datos históricos, búsquedas en Google Trends en Brasil y datos climáticos.

En esta investigación, como solución al problema se pretende elaborar un modelo de Machine Learning, empleando el algoritmo supervisado Extreme Gradient Boosting (XGBoost) basado en árboles de decisión para la predicción de la incidencia del dengue, aplicable en la región San Martín, con el propósito de demostrar la importancia de los datos de fuentes alternativas fiables, revalorar el uso de las herramientas tecnológicas y elaborar un modelo de predicción para afrontar y prever la propagación de la epidemia, que contribuya en la toma de decisiones para la realización de las acciones preventivas y de control del dengue.

Por consiguiente, la problemática investigada en el trabajo fue: ¿Será posible predecir la incidencia del dengue empleando un modelo de aprendizaje supervisado basado en el algoritmo extreme gradient boosting (XGBoost)?; con el propósito de responder la incógnita se planteó la hipótesis: Mediante un modelo de aprendizaje supervisado basado en el algoritmo extreme gradient boosting (XGBoost) se podrá predecir la incidencia del dengue.

Lo que respecta a los objetivos, se estableció como objetivo general: Predecir la incidencia del dengue utilizando un modelo de aprendizaje supervisado basado en el algoritmo extreme gradient boosting (XGBoost); y los específicos: 1) Elaborar un modelo de aprendizaje supervisado empleando el algoritmo paralelizable extreme gradient boosting (XGBoost) basado en árboles de decisión; 2) Utilizar datos de fuentes alternativas fiables para afianzar la data histórica existente y garantizar mayor precisión de la predicción; 3) Construir una herramienta tecnológica basado en dashboard para mostrar los indicadores de los modelos en la predicción de la incidencia del dengue.

CAPÍTULO II
MARCO TEÓRICO

2.1. Antecedentes de la investigación

En el estudio de (Koplewitz et al., 2022) su objetivo fue estimar la incidencia del dengue a nivel de ciudades e identificar el grado de contribución de las diferentes fuentes de datos en los modelos de predicción. Emplearon el algoritmo de aprendizaje supervisado basado árboles (Random Forest) y métodos de regresión lineal (LASSO) para pronosticar la incidencia del dengue empleando fuentes datos como data histórica, Google Trends y del clima. Sus resultados evidencian que el modelo de aprendizaje supervisado (Random Forest) tuvo una precisión de RMSE 11.047 frente al 12.485 del método de regresión lineal (LASSO), en R-RMSE 0.354 y 0.4 respectivamente, y los predictores con mejor desempeño fueron los datos históricos combinado con el de Google Trends.

En su artículo (Benedum et al., 2020) predicen brotes y recuentos semanales del dengue en lugares endémicos. Comparan algoritmos de aprendizaje automático supervisado (Random Forest y Random Forest – Univariate Flagging Algorithm), modelos de regresión (regresión de Poisson y regresión logística) y modelos de series de tiempo (ARIMA) junto a una combinación de fuentes de datos de los lugares estudiados. Sus resultados revelan que los modelos de aprendizaje automático supervisado tienen una mejor precisión predictiva de la incidencia del dengue que otros tipos de modelos siempre y cuando se combine las fuentes de datos.

En su investigación (Zhao et al., 2020) evaluan la capacidad predictiva de los modelos de aprendizaje automático en la predicción de los casos de dengue. Utilizan el algoritmo de aprendizaje supervisado (Random Forest), una red neuronal artificial (ANN) y modelos de series de tiempo (ARIMA), junto a una combinación de diferentes fuentes de datos en periodos semanales. Sus resultados demuestran que el modelo con mejor desempeño fue el basado en aprendizaje automático supervisado (Random Forest), teniendo mayor precisión en el pronóstico de la incidencia del dengue, con una perspectiva viable a corto y largo plazo.

En su artículo de investigación (Souza et al., 2022) mejoran la predicción de brotes de dengue en Brasil utilizando datos baratos de fuentes alternativas para aumentar la probabilidad de adopción por parte de las autoridades públicas. Emplean técnicas de computación moderna de procesamiento de datos y de aprendizaje múltiple como el modelo máquina de vector de soporte (SVM) basado en aprendizaje automático. Los

resultados demuestran que se mejoró la precisión de la predicción de un 0.72 a 0.80, afianzando una forma de predicción de casos de dengue en centros urbanos.

En la investigación de (Aiken et al., 2020) pronostican la incidencia de enfermedades endémicas empleando métodos epidemiológicos digitales. Aplican modelos de aprendizaje automático, principalmente autorregresión dinámica multivariante para generar estimaciones de brotes y la utilización de datos históricos junto a volúmenes de consulta de Google Trends en relación a cada enfermedad. Evidencian que los modelos de aprendizaje automático realizaron predicciones de los brotes de las enfermedades y el mejor desempeño fueron los entrenados con datos históricos combinados con los de Google Trends.

En el artículo de (Mussumeci & Codeço Coelho, 2020) predicen la incidencia del dengue para la realización de medidas de control en una temporada elevada de transmisión. Comparan modelos de aprendizaje automático supervisado (Random Forest), métodos de regresión (Lasso) y una red neuronal profunda (LSTM), siendo entrenados con datos combinados provenientes de múltiples fuentes. Sus resultados develan que LSTM tuvo mayor precisión predictiva para pronosticar brotes del dengue, pero en la aplicación práctica es demasiado costoso implementarlo, siendo Random Forest y Lasso menos costosos en términos computacionales y altas probabilidades de aplicabilidad para organizaciones públicas encargadas del manejo de la epidemia.

2.2. Fundamentos teóricos

2.2.1. Algoritmos de aprendizaje automático (ML)

Los algoritmos de aprendizaje automático son una serie de instrucciones que la máquina debe realizar a fin de lograr un objetivo que generalmente están centrados en la detección o establecimiento de patrones y/o características para realizar predicciones o clasificaciones teniendo en cuenta el tiempo y la eficiencia del desempeño, emplean datos de entrenamiento para aprender de la actividad a efectuar, mientras mayor sea la cantidad de datos de entrenamiento, el desempeño del modelo frente al problema tendrá una representación más cercana a la realidad y lo evidencia con resultados precisos (Ferdous et al., 2020; Latif et al., 2019).

El aprendizaje automático es interdisciplinario, destacan en aplicaciones como la computación, medicina e ingeniería, debido a que, ayudan a resolver incógnitas complejas que por métodos manuales o tradicionales se imposibilita, desde un enfoque económico, reduce hasta un 25% los costos de operaciones, permitiendo optimizar el aprovechamiento de los recursos, debido a esto su popularidad va en

aumento a causa del desarrollo de componentes tecnológicos relacionado al procesamiento de datos y al mayor interés en el área por parte de la comunidad empresarial y científica, (Khan et al., 2022; Lee & Shin, 2020; Portugal et al., 2018). Los algoritmos de aprendizaje automático generalmente son utilizados para problemas de clasificación, regresión, cambios de valor en el tiempo, detección de anomalías y detección de similitudes. Los modelos de aprendizaje automático brindan herramientas potentes para la resolución de problemas particulares y permite abordarlos correctamente, debido a la digitalización y la creciente utilización de la tecnología, la disponibilidad de datos es cada vez mayor y los algoritmos de aprendizaje automático destacan sobre los comúnmente utilizados enfoques estadísticos (Bi et al., 2019).

Algunos ejemplos específicos (Ray, 2019):

- Estimación del precio en una hora pico de tráfico en la app de Uber.
- Chatbots para la atención personalizada al cliente.
- Refinación de resultados del motor de búsqueda de Google buscador.
- Reconocimiento facial en la red social Facebook.
- El asistente personal de Google.
- Recomendaciones de productos de las tiendas en línea.
- Publicidad personalizada.
- Predicción de estafas en línea.
- Filtrado del correo basura en Google Gmail.
- Detección del cáncer en relación al historial médico del paciente.

Tipos de algoritmos de aprendizaje automático (ML)

A fin de emplear el algoritmo de aprendizaje automático adecuado, es necesario conocer el tipo de problema al que trata de ofrecer solución, sea de agrupamiento, regresión o clasificación siempre y cuando estén disponibles conjuntos de datos que permita entrenar el modelo, estos deben tener las características necesarias que para abordar el inconveniente, y a su vez, los tipos y la clase de datos determinarán la técnica a utilizar, que puede ser aprendizaje automático supervisado, no supervisado, semisupervisado o de refuerzo. Los algoritmos de aprendizaje automático más populares son (Ray, 2019):

- **Descenso del gradiente**

Es un procedimiento reiterativo que busca minimizar la función de costo, calculando la derivada respecto a una de las variables en función al gradiente o pendiente. El cálculo de los coeficientes se da en cada repetición empleando el negativo de la

derivada y la tasa de aprendizaje reduce los coeficientes en cada paso, esto se da a fin de conseguir los mínimos locales y las iteraciones culminan al no existir reducción de la función de costo, es decir, se alcanzó el valor mínimo de esta. Existen tres métodos de este tipo (Ray, 2019):

A. Descenso de gradiente estocástico – estocástico aleatorio (SGD)

Calcula el error en cada ejemplo de entrenamiento en relación conjunto de datos y actualiza los parámetros para cada ejemplo. Tiene la fortaleza de que las actualizaciones evidencia una tasa de mejora; sin embargo, es costoso computacionalmente en comparación con el BGD (Ray, 2019).

B. Descenso de gradiente por lotes (BGD)

Se calcula el error en cada ejemplo que contiene el conjunto de datos de entrenamiento. Destaca en su eficiencia computacional, pero flaquece en que el gradiente del error puede dar como resultado una convergencia inadecuada (Ray, 2019).

C. Descenso de gradiente de lotes pequeños (MBGD)

Es la combinación del SGD y BGD, el conjunto de datos es dividido en lotes pequeños y se actualiza a cada uno. Es el punto intermedio de la eficiencia y precisión, se utiliza principalmente en aprendizaje profundo para el entrenamiento de una red neuronal, la desventaja más notoria es al configurar la tasa de aprendizaje que, de realizarlo mal, omitirá el mínimo local o nunca convergerá (Ray, 2019).

- **Regresión lineal**

Está orientado al aprendizaje supervisado, utilizado para predicciones y modelado de variables continuas. La regresión emplea grupos de datos con etiquetas, el valor de las variables de entrada determina el valor de las de salida, una representación simple de este tipo de algoritmo es el ajuste de un hiperplano recto a un determinado conjunto de datos y la relación de estas es lineal. Este algoritmo destaca por la facilidad de comprender su funcionamiento y de evitar el sobreajustamiento; sin embargo, simplifica demasiado el problema abstraído haciéndola poca recomendada para utilizar en problemas del mundo real debido a que generalmente estos tienen tendencia no lineal, o cuando existen patrones complejos en el conjunto de datos (Ray, 2019).

- **Regresión multivariante**

Los problemas existentes en el mundo real son complejos pues la regresión lineal no se ajusta correctamente a estos, esta situación se puede manipular desde otra perspectiva, que una variable dependa de muchas características, un claro ejemplo es el valor de una vivienda, que el precio está en relación a la zona de la residencia, el tamaño del terreno, el valor del metro cuadrado, el tipo de zona, etc. Estas dimensiones pueden permitir tener un mejor modelo por la relación entrada-predictor y la dependencia de la salida-respuesta, pero no necesariamente el adicionar más entradas se refleje positivamente en la precisión de la predicción hasta puede llegar a pasar todo lo contario y perjudicar el rendimiento del modelo, el caso más óptimo al emplear este algoritmo sería que las variables entrantes posean relación con las de salida. La gran desventaja del algoritmo radica en la complejidad, el requerimiento de experiencia y conocimientos en técnicas y modelado estadístico (Ray, 2019).

- **Regresión logística**

Óptimo para problemas de clasificación, ofrece el resultado como la probabilidad de suceso de un evento en escala de 0 a 1, en relación a las entradas, se puede trabajar de manera binomial (expresado en dos términos) o multinomial (más de dos términos), es decir, es empleado para predecir la variable objetivo categórica. La ventaja del algoritmo está en la facilidad de la implementación, buen rendimiento computacional y de entrenamiento, simplicidad para regularizar, sin escalado de las variables de entrada, no le afecta el ruido de la data o la multicolinealidad; pero, sufre bastante para tratar inconvenientes no lineales y puede sobreajustarse fácilmente (Ray, 2019).

- **Árbol de decisión**

Con enfoque en el aprendizaje automático supervisado, aborda problemas de regresión y clasificación dividiendo continuamente los datos en relación de un determinado parámetro, las denominadas hojas son las decisiones del modelo y los nodos a los datos. En el árbol de clasificación, las decisiones tienen el funcionamiento de la forma sí o no, es decir, categórica, y en el árbol de regresión, las decisiones son continuas. Este algoritmo tiene la fortaleza de ser fácilmente interpretado, buen rendimiento, completar datos faltantes en relación a probabilidades, manipulación de variables cuantitativas y cualitativas. La desventaja de los árboles de decisión radica en lo vulnerable que es al sobreajuste; no obstante, la solución para esta circunstancia puede ser el empleo de Random Forest por estar basado en el modelado de conjuntos, los problemas recurrentes del algoritmo son el control del tamaño del árbol

(puede ser solucionado con la “poda”) y tendencia de ocurrencia de errores en el muestreo (Ray, 2019).

- **Máquina de vectores de soporte (SVM)**

Maneja problemas de clasificación y regresión para ello, se requiere definir el límite de la decisión (hiperplano), ya que, al existir un conjunto de datos estos para ser separados necesitan un plano de decisión, los objetos pueden o no ser separados linealmente, en la cual se necesitaría expresiones matemáticas complicadas (denominada núcleos) a fin de separarlos, el objetivo del algoritmo es clasificar en función de los datos de entrenamiento, el sobreajustamiento se puede corregir con generalización, se puede escalar datos con alta dimensión; a pesar de ello, sufren contra gran cantidad de datos, padece con datos ruidosos y son difícilmente comprensibles (Ray, 2019).

- **Aprendizaje bayesiano**

Elige una distribución de probabilidad anterior y va actualizando para conseguir una posterior y con observaciones obtenidas a su vez puede ser empleada como anterior, manipula relativamente bien la ausencia de datos, evita sobreajustamiento de los datos, no hay necesidad de eliminar contradicciones en la data; pero, las falencias más importantes son la dificultad para la elección previa, la influencia de la distribución posterior por la anterior perjudicando el rendimiento de la predicción, y siendo computacionalmente costoso (Ray, 2019).

- **Ingenuo Bayes**

Basado en la probabilidad condicional, mediante una tabla de probabilidad (modelo), es actualizado por los datos de entrenamiento, basado en características de los valores a fin de encontrar probabilidades para la predicción. La independencia condicional le da el nombre de “ingenuo”, de fácil implementación, buen rendimiento, sin requerir muchos datos, trabaja con datos continuos o discretos, funciona con la clasificación binaria o múltiple; sin embargo, la simplicidad del modelo puede influir en el desempeño, computacionalmente costos de existir muchas variables (Ray, 2019).

- **Vecino k más próximo (KNN)**

El algoritmo maneja problemas de clasificación, empleando puntos de datos en una base de datos, en base a ello, clasifica en relación al punto de datos. Es no paramétrico por no asumir distribuciones de datos subyacentes, es un método relativamente simple y de fácil implementación, flexible y se adecua a clases

multimodales, maneja etiquetas de clase; no obstante, es costoso computacionalmente, las características ruidosas influyen en la precisión (Ray, 2019).

- **Propagación hacia atrás**

Brinda una manera eficiente y simple para el cálculo del gradiente de una red neuronal y utilización del descenso de gradiente estocástico (SGD). Empleado en el aprendizaje profundo. Las redes neuronales (NN) tienen un buen desempeño ante escenarios sin criterios o reglas definidas para dar con la respuesta; a pesar de ello, es de complicada explicación la manera a la que se llegó a la solución. El algoritmo al ser empleado en una red neuronal artificial (ANN) con capas ocultas, aumenta el costo computacional, dificultad en la convergencia y mínimos locales (Ray, 2019).

Aprendizaje automático supervisado

Es una actividad del aprendizaje automático para aprender una función que establece una entrada en relación a una salida (par de entrada-salida), la deducción se realiza debido al conjunto de datos etiquetados a ser empleados en el entrenamiento del modelo, entonces los algoritmos de aprendizaje automático supervisado requieren de asistencia exterior. Los datos a utilizar deben ser fraccionados en datos de entrenamientos y de prueba, donde el primero tendrá una variable de salida que será para clasificar o predecir (según sea el caso). Los algoritmos tienen que aprender los patrones de los datos de entrenamiento para posterior a este, pueda ser validado por los datos de las pruebas. Este aprendizaje es común en redes neuronales y algoritmos basados en el árbol de decisión. Este tipo de aprendizaje automático destaca en tareas de regresión y clasificación (Muhammad & Yan, 2015; Nasteski, 2017).

Extreme gradient boosting (XGboost)

XGBoost (Extreme gradient boosting) es un algoritmo basado en árboles de decisión potenciados por gradientes, esta es una técnica de aprendizaje automático, produciendo un modelo para predicción que está conformado por grupos de modelos de pronóstico débiles para conformar uno poderoso, a fin de aumentar la precisión del modelo, se emplea la ingeniería de características o aplicando algoritmos de impulso inmediato (el caso de XGBoost). El algoritmo ofrece paralelización en la elaboración del árbol de decisiones al realizar el entrenamiento y la optimización del caché para la estructura de datos. Las características más resaltantes de XGBoost, las razones más importantes para elegir a este algoritmo es la velocidad de ejecución y buen desempeño del modelo (Mitchell & Frank, 2017).

Los pasos para implementar el algoritmo de XGBoost para la mayoría de casos son:

1. Cargar todas las librerías necesarias.
2. Cargar data de entrenamiento y prueba.
3. Cargar etiquetas para los datos de entrenamiento.
4. Combinar los datos de entrenamiento y prueba.
5. Limpieza de variables.
6. Preprocesamiento de datos empleando características categóricas.
7. Separar los datos de entrenamiento y prueba previamente combinados.
8. Configurar los parámetros generales.
9. Configurar los parámetros de potenciación.
10. Configurar los parámetros específicos para potenciación lineal.
11. Configurar parámetros de aprendizaje
12. Entrenar el modelo.
13. Solicitar valores de predicción con los datos de prueba.

2.2.2. Dengue

Es una enfermedad del tipo arbovirus que tiene como vector de transmisión al mosquito Aedes aegypti, un artrópodo que causa 390 millones de infecciones anualmente, 100 millones aproximadamente son asintomáticos y 10 mil decesos en más o menos 125 países. Se conjetura que el virus aparece en un lapso donde los huéspedes eran primates no humanos y estos por un ciclo epidémico/endémico, infectó al ser humano hace más de 1000 años (Ratanakomol et al., 2022; Sanchez-Gendriz et al., 2022).

Controlar al vector reduciendo la densidad poblacional de estos, es una táctica significativa para la prevención y control de las epidemias transmisibles. El desarrollo de las larvas del mosquito que generalmente aparecen en recipientes u objetos que funjan de depósitos de agua con el propósito de eliminar los posibles hábitats de reproducción del vector, funcionando como acción preventiva; sin embargo, no perjudica directamente a la prevalencia de la enfermedad por mosquitos adultos, durante un brote de esta, la opción más viable es emplear insecticidas para fumigar espacios, siendo una medida de control apropiada para brotes focalizados y con buenos resultados para reducir la transmisibilidad del dengue, sumado a la educación de la comunidad y la participación de esta, no garantiza el éxito e incluso la sostenibilidad tiende a ser desafiante, ocasionando el incremento del impacto en la salud pública de países endémicos con recursos limitados. (Lenhart et al., 2022; Zheng et al., 2022).

Predicción del dengue

La estimación de la carga del dengue es difícil de realizar por los retrasos para identificar casos positivos, las variaciones anuales en la incidencia, casos asintomáticos, carencia de recursos, dificultad para emplear eficientemente los datos, causando que sea desafiante la implementación de medidas de contingencia para lidiar con brotes de la enfermedad. Estos problemas pueden ser superados con el desarrollo de herramientas de alerta temprana con la capacidad de predecir la incidencia de la enfermedad.

Existen varios enfoques de modelado de alertas tempranas, siendo el aprendizaje automático el más exitoso siempre y cuando haya disponibilidad de datos para abordar el problema; sin embargo, el modelo puede ser difícil de parametrizar y las suposiciones de este pueden no ser claras hasta posterior al brote, a causa de ello, el enfoque de conjuntos (Árbol de decisión, Random Forest, XGboost, etc) con métodos de pronóstico, tienen un buen desenvolvimiento y ha crecido en popularidad debido al buen rendimiento y precisión que presenta, esta última es importante para evaluar el desempeño del modelo utilizado en la problemática (Benedum et al., 2020). Algunas de las métricas de evaluación de precisión son:

- **Error cuadrático medio (RMSE)**

Es la raíz cuadrada de la diferencia al cuadrado de valores pronosticados y valores reales entre la cantidad de valores, es decir, es la desviación estándar del error de predicción, de utilidad para comparar la precisión de la predicción sobre un conjunto de datos en particular. Esta medida es no negativa y cuando es 0, significa que el modelo se ajusta perfectamente a los datos, generalizando, si el RMSE de un modelo es menor a otro, tiene mejor precisión (Hyndman & Koehler, 2006).

- **Error relativo cuadrático medio (R-RMSE)**

Este indicador es calculado dividiendo al RMSE con los datos pronosticados, puede estar expresado entre 0 y 1, o multiplicarse por 100 para ser representado en porcentaje (Despotovic et al., 2016).

- **Coeficiente de determinación (R^2)**

Definida como la proporción de la varianza total de la variable expuesta en la regresión, evidencia la bondad del ajuste de la variable a explicar en un modelo. El indicador oscila entre 0 y 1, mientras más cerca esté del 0 menos ajustado está el modelo, y de lo contrario si tiende al 1 (Rodríguez, 2005).

- **Coeficiente de correlación de Pearson**

Cuantifica la asociación de dos variables numéricas, denominada como la covarianza estandarizada, la relación a estudiar debe ser lineal o podría presentar problemas, puede asumir valores entre -1 a 1, siendo 1 una relación positiva perfecta, -1 una relación negativa perfecta y con tendencia a 0 que no existe asociación entre las dos variables (Lalinde et al., 2018).

2.2.3. Definición de términos básicos

Algoritmo: son procedimientos o reglas definidas, ordenadas, lógicas y finitas para producir una salida o resultado (Shanker, 1987).

Aprendizaje automático: Se les ofrece a las computadoras la capacidad de aprender sin estar explícitamente programadas generalizando el inconveniente a afrontar (Burrell, 2016; Sarker, 2021).

Costo computacional: Es el costo que implica un sistema de cómputo para realizar una determinada actividad, resolver una situación o cálculo matemático, se da a nivel de hardware y software.

Covarianza: Es la variación de dos variables aleatorias en relación a las medias de estas para comprobar dependencia entre las variables, útil a fin de estimar la recta de regresión o el coeficiente de correlación lineal.

Hiperparámetro: Son valores empleados a fin de configurar el modelo de aprendizaje automático en la etapa de entrenamiento (Feurer & Hutter, 2019).

Inteligencia artificial: Definido como la aplicación de múltiples algoritmos para que una computadora pueda emular la inteligencia humana en la realización de una determinada actividad (Corvalán, 2018).

Parámetro: Son valores obtenidos posterior a la fase de entrenamiento de un modelo de aprendizaje automático, útil para hacer predicciones y precisar la capacidad de un modelo frente a un inconveniente.

Precisión: es una métrica que evidencia el desempeño del modelo de clasificación frente a la resolución de un problema, en relación de las predicciones correctas y el número total de predicciones (Fleuren et al., 2020).

Predicción: Salidas de un modelo de aprendizaje automático posterior a ser entrenado por un grupo de datos; es decir, calcula un probable resultado a futuro (Fleuren et al., 2020).

CAPÍTULO III
MATERIALES Y MÉTODOS

3.1. Ámbito y condiciones de la investigación

3.1.1 Contexto de la investigación

Pretendemos ser una base de estudio en relación a la problemática identificada en la región San Martín, Perú; mediante las predicciones de un modelo de aprendizaje automático supervisado.

3.1.2 Periodo de ejecución

La ejecución del proyecto se realizó durante 4 meses, de diciembre del 2022 a marzo del 2023.

3.1.3 Autorizaciones y permisos

La presenta investigación empleó conjuntos de datos de repositorios abiertos sea el caso del estudio de (Koplewitz et al., 2022), pero no hubo implicancia de restricción alguna de acceso o manipulación de materiales, debido a que no se empleó sustancias o reactivos prohibidos por alguna norma a nivel nacional, regional o local; es decir, no aplica.

3.1.4 Control ambiental y protocolos de bioseguridad

El proyecto no estuvo involucrado en manejo biológico o ambiental, debido a la nueva normalidad a causa del Covid-19, el desarrollo del estudio no requirió demasiada interacción física con las personas involucradas en la problemática y en la solución; en otras palabras, no aplica el control ambiental y/o las medidas de bioseguridad puesto que no puso en riesgo a los involucrados en el desarrollo de la investigación.

3.1.5 Aplicación de principios éticos internacionales

Los investigadores hacen mención que su intervención respetó los principios éticos generales de la investigación; particularmente la originalidad, ya que no se fabricaron o falsificaron los datos, y se citaron las fuentes consultadas. También se tuvo en cuenta los aspectos éticos de consentimiento informado al personal encuestado, de este modo se aseguró la práctica investigativa forjada en buenos valores que viabilicen los resultados del estudio.

3.2. Sistema de variables

3.2.1 Variables principales

La operacionalización de las variables es de la siguiente forma:

Tabla 1

Descripción de variables por objetivo específico

Objetivo general: Predecir la incidencia del dengue utilizando un modelo de aprendizaje supervisado basado en el algoritmo extreme gradient boosting (XGBoost).

Variable abstracta	Variable concreta	Medio de registro	Unidad de medida
Predicción de la incidencia del dengue.	Error cuadrático medio (RMSE)	Informe	Numérico
	Error cuadrático medio relativo (R-RMSE)	Informe	Numérico
	Coeficiente de determinación (R^2)	Informe	Numérico
	Coeficiente de correlación de Pearson	Informe	Numérico

Fuente: Elaboración propia.

3.2.2 Variables secundarias

No aplica.

3.3 Procedimientos de la investigación

a) Tipo y nivel de investigación

La investigación fue de tipo aplicada, debido a que empleamos conocimientos en aprendizaje automático supervisado e inteligencia artificial a fin de elaborar un modelo de predicción utilizando el algoritmo de extreme gradient boosting (XGboost) para el pronóstico de la incidencia del dengue. De nivel descriptivo, ya que pretendimos describir la precisión de la predicción de la incidencia del dengue empleando diferentes fuentes de conjuntos de datos a fin de generar una línea de base de estudio de la problemática para próximas investigaciones

El diseño de la investigación es no experimental porque fue encaminada a observar e identificar el mejor modelo entrenado con las fuentes de datos (incluyendo la combinación de estas) a fin de pronosticar los brotes del dengue empleando el

algoritmo extreme gradient boosting (XGBoost), no existió control alguno en las variables, por ello la mejor representación es:

$$M_{jn} \longrightarrow O_D$$

Donde, M fue el modelo entrenado, j las ciudades (20 en total) que conformaron el set de datos, n las fuentes de datos empleadas siendo, n = 1 la aplicación de datos históricos, n = 2 el uso de la data obtenida de Google Trends y, n = 3 la combinación de la data histórica y de Google Trends, O_D hace referencia a la observación, análisis y descripción de las salidas del modelo construido usando XGBoost para predecir la incidencia del dengue.

b) Población y muestra

La población estuvo conformada por 345 periodos semanales de datos de casos de dengue de diferentes ciudades de Brasil, obtenidos directamente del Ministerio de Salud del mismo país, los datos meteorológicos son de acceso público y pertenecen a la Oficina de Asimilación y Modelado Global (GMAO) del Centro de Vuelo Espacial Goddard de la NASA (CVEG), y los datos de Google Trends fueron extraídos por la herramienta “GTrends-Tools” en relación a busquedas que tengan relación al dengue, así mismo los datos estaban correlacionados en periodos semanales con fechas del 03-01-2010 al 31-07-2016, estos datos se obtuvieron del dataset empleado por (Koplewitz et al., 2022) con repositorio en https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/QB4C7N.
La muestra fue de 292 periodos semanales debido a que en el preprocesamiento vimos conveniente reducir estos para mejorar los resultados de la predicción siendo ahora fechas del 02-01-2011 al 31-07-2016, por ello el muestreo será no probabilístico.

c) Diseño analítico, muestral y experimental

La investigación fue de diseño no experimental, descriptivo y observacional.

En la realización, empleamos un conjunto de datos de 345 periodos semanales de data de casos dengue, de búsquedas en Google en relación al dengue (visibles en Google Trends) y datos climatológicos, que posteriormente en el preprocesamiento de los datos, se redujo a 292 periodos semanales empleados en la investigación de (Koplewitz et al., 2022).

Construimos varios modelos de aprendizaje automático supervisado empleando el algoritmo extreme gradient boosting (XGBoost) que fueron entrenados en relación a 20 ciudades de Brasil (Ji-Paraná, Manaus, São Luís, Parnaíba, Juazeiro do Norte,

Maranguape, São Vicente, Sertãozinho, Santa Cruz do Capibaribe, Aracaju, Eunápolis, Belo Horizonte, Barra Mansa, Rio de Janeiro, São Gonçalo, Barretos, Barueri, Guarujá, Três Lagoas y Rondonópolis) y grupos de datos en relación al dengue (el número de semana según la ISO-8601 y datos climatológicos, búsquedas de Google Trends y una fusión de ambas).

De los modelos entrenados en relación a las ciudades y el conjunto de datos correspondiente, obtuvimos el error cuadrático medio (RMSE), error cuadrático medio relativo (R-RMSE), coeficiente de determinación (R2) y el coeficiente de correlación de Pearson a fin de determinar la precisión del modelo en el pronóstico de brotes de dengue.

Estos datos fueron registrados en una base de datos para construir un dashboard a fin de evidenciar la predicción del dengue en relación al desempeño de los modelos entrenados empleando el algoritmo XGBoost.

d) Representación de la información

La recolección de datos lo realizamos empleando el algoritmo adjuntado en el Anexo 2, en un archivo Excel que contiene la información de los indicadores de cada modelo generado en relación a las ciudades y las fuentes de datos, para posteriormente ser cargado en una base de datos Mysql para consumirse empleando el software Power BI versión 2022 para la generación del dashboard.

e) Análisis estadístico

Para contestar la hipótesis de este trabajo, al realizar la comparación de los indicadores en relación a las ciudades y fuentes de datos, se verificó qué modelos se ajustan de mejor y peor manera teniendo en conocimiento la teoría y los parámetros que competen a cada indicador (RMSE, R-RMSE, R^2 y Correlación de Pearson) mencionado en el subacápite 2.2.3.

3.3.1 Objetivo específico 1:

Elaborar un modelo de aprendizaje supervisado empleando el algoritmo paralelizable extreme gradient boosting (XGBoost) basado en árboles de decisión.

Las actividades del objetivo se explican según el diagrama:

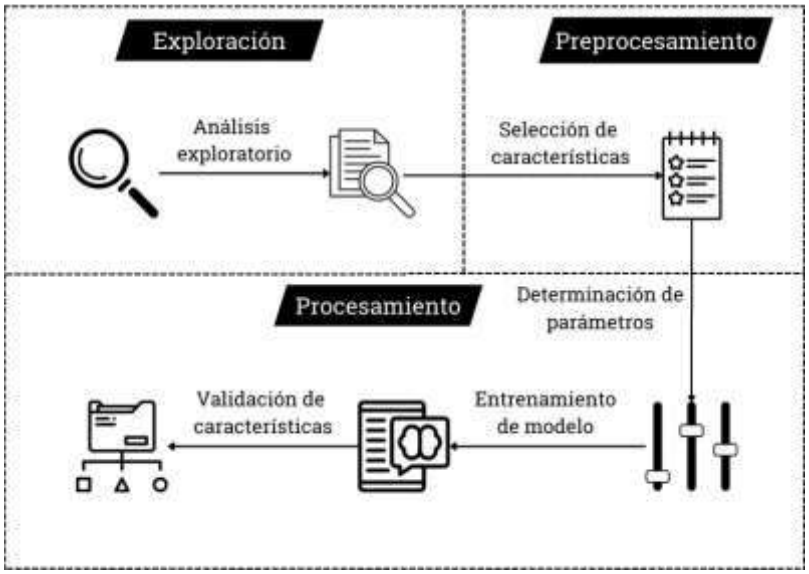


Figura 1
Diagrama de pasos del objetivo específico 1

a) Exploración:

De este paso en adelante, utilizamos la herramienta Jupyter Notebook, específicamente la que posee interfaz web de versión 6.4.12 de código abierto y el lenguaje de programación Python 3.10.8.

- **Análisis exploratorio.**

Partimos explorando la conformación de la data el set de datos, considerando aspectos importantes como que el archivo tenía extensión ".xlsx" característico en hojas de cálculo de Microsoft Excel. Al aperturarse el archivo, se observó que contiene varias hojas de cálculo con datos importantes, la primera hoja por defecto llamada "Dengue", poseía datos históricos segmentados por semanas, teniendo datos desde el 03/01/2010 al 31/07/2016.

Vimos conveniente conocer el tipo de dato que contenía cada columna, de la hoja inicial, se importó los del archivo "Dengue_selected_cities.xlsx", a la variable df empleando la librería de pandas:

```
import pandas as pd
df = pd.read_excel(r'Dengue_selected_cities.xlsx').dtypes
```

La variable “df” contenía la información de:

Tabla 2

Hoja principal, columnas por tipo de dato

Columna	Tipo de dato
Date	datetime64
110012 Ji-Paraná	int64
130260 Manaus	int64
211130 São Luís	int64
220770 Parnaíba	int64
230730 Juazeiro do Norte	int64
230770 Maranguape	int64
241300 São Vicente	int64
251593 Sertãozinho	int64
261250 Santa Cruz do Capibaribe	int64
280030 Aracaju	int64
291072 Eunápolis	int64
310620 Belo Horizonte	int64
330040 Barra Mansa	int64
330455 Rio de Janeiro	int64
330490 São Gonçalo	int64
350550 Barretos	int64
350570 Barueri	int64
351870 Guarujá	int64
500830 Três Lagoas	int64
510760 Rondonópolis	int64

Revisamos observacionalmente la hoja de Excel en relación a los valores por fechas y ciudad, a lo largo del 2010 en varias ciudades encontramos que no había registros de dengue, es decir, tenían como valor “0” dada una fecha, para corroborarlo, empleamos el siguiente código:

```
import pandas as pd

# Cargar el archivo Excel
df = pd.read_excel(r'Dengue_selected_cities.xlsx')
df = df.loc[ df["date"] < "2011-01-01"]
column_names = df.columns.tolist()
column_names = column_names[1:]

# Crear un nuevo dataframe para almacenar los resultados
result_df = pd.DataFrame()

# Iterar sobre las columnas
for column_name in column_names:

    # Contar la cantidad de ceros por fila para la columna actual
    zeros_count = (df[column_name] == 0).sum()

    # Agregar una nueva fila al resultado dataframe
    result_df = result_df.append({'Columna': column_name,
'Cantidad de Ceros': zeros_count}, ignore_index=True)
```

La variable “result_df” contenía:

Tabla 3
Cantidad de ceros por ciudad

Columna	Cantidad de ceros
110012 Ji-Paraná	32
130260 Manaus	0
211130 São Luís	6
220770 Parnaíba	42
230730 Juazeiro do Norte	12
230770 Maranguape	39
241300 São Vicente	52
251593 Sertãozinho	34
261250 Santa Cruz do Capibaribe	29
280030 Aracaju	7
291072 Eunápolis	41
310620 Belo Horizonte	0
330040 Barra Mansa	22
330455 Rio de Janeiro	0
330490 São Gonçalo	1
350550 Barretos	5
350570 Barueri	27
351870 Guarujá	24
500830 Três Lagoas	18
510760 Rondonópolis	17

Considerando que el año 2010 tiene 52 semanas, varias ciudades como São Vicente, Parnaíba, Maranguape entre otros, poseen altas cantidades registros de valores “0” en estas fechas, por ello, para posteriores procedimientos se empleará data dentro del 2011, con el propósito de emplear data afianzada y uniformizar los modelos en relación a cada ciudad.

b) Preprocesamiento:

- **Selección de características.**

Las otras hojas del set de datos, tenían el nombre de cada una de las 20 ciudades, y a su vez contenía columnas y datos, el nombre de las columnas se repetía en cada hoja en relación a la ciudad, para poder observarse tomamos como ejemplo la ciudad “Manaus”, hicimos uso del siguiente código:

```
import pandas as pd
df = pd.read_excel(r'Dengue_selected_cities.xlsx',
                   sheet_name = "Manaus").dtypes
```

La variable “df” contenía:

Tabla 4

Columnas por tipo de dato en cada ciudad

Columna	Tipo de dato
date	datetime64
N_cases	int64
dengue	float64
a dengue	float64
sintomas da dengue	float64
sintomas dengue	float64
sobre a dengue	float64
sintomas de dengue	float64
Mosquito da dengue	float64
Mosquito Dengue	float64
dengue hemorragica	float64
mosquito	float64
mosquitos	float64
temp	float64
wind	float64
percipitation	float64
humidity	float64

Según (Koplewitz et al., 2022), la columna “date” y “N_cases”, corresponden a las fechas y la cantidad de casos de dengue, las columnas “dengue”, “a dengue”, “sintomas da dengue”, “sintomas dengue”, “sobre a dengue”, “sintomas de dengue”,

“Mosquito da dengue”, “Mosquito Dengue”, “dengue hemorrágica”, “mosquito” y “mosquitos”, corresponden a las palabras clave recopiladas de Google Trends mediante la herramienta “GTrends-Tools”, que multiplica una constante anónima en los datos recopilados para realizar una normalización relativa en estos, se realiza este procedimiento debido al propio funcionamiento de Google Trends y las políticas propias de esta en vislumbrar la data; las otras columnas “temp”, “wind”, “percipitation” y “humidity”, corresponden a datos meteorológicos, dada la perspectiva de esta investigación, no se emplearon.

Decidimos hacer uso de la data histórica y las fechas en relación a cada hoja del set de datos correspondiente a cada ciudad, para la creación de este modelo, el punto de partida fue una ciudad de ejemplo, “Maranguape” para este caso, adicionalmente se hizo uso de diferentes librerías para leer, manipular y representar los datos, estos fueron:

- Pandas, de utilidad para la manipulación y el análisis de los datos con una perspectiva tabular.
- Numpy, permitió el eficiente uso de operaciones numéricas en vectores y matrices.
- Matploit es una librería centrada en la visualización de gráficos.
- Seaborn, una librería que permitió vislumbrar ciertos tipos de gráficos.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Seleccionamos los datos en relación a la ciudad y columnas a emplear, en este caso data histórica en relación al tiempo.

```
dataframe = pd.read_excel(r'Dengue_selected_cities.xlsx',
sheet_name = "Maranguape", usecols = ["date", "N_cases", "temp",
"wind", "percipitation", "humidity"])
```

Establecimos la fecha como índice para posteriormente, seleccionar datos solo a partir del 2011 en adelante.

```
dataframe = dataframe.set_index("date")
dataframe = dataframe.loc[ dataframe.index > "2011-01-01"]
```

Graficamos los casos de dengues semanales, para ello se utilizó:

```
dataframe["N_cases"].plot(style=".", figsize=(15, 5),
color=paleta_colores[0], title="Casos de dengue semanales")
```

Siendo la salida del código el siguiente gráfico:

Figura 2
Casos de dengue semanales.

Procedimos con la generación de una característica nueva en relación a la fecha previamente establecida como índice, de tal manera, se obtuvo una nueva columna llamada “week”, esta es el número de la semana que corresponde determinada fecha, que pretende correlacionarse en relación al número de casos, empleando lo siguiente:

```
def crear_caracteristicas(dataframe):
    dataframe["week"] = dataframe.index.isocalendar().week
    dataframe["week"] = dataframe["week"].astype(np.int64)
    return dataframe
dataframe = crear_caracteristicas(dataframe)
```

Realizamos un nuevo gráfico, que permitió observar el número de casos y como se distribuye de manera visual en relación a la semana.

```
fig, ax = plt.subplots(figsize = (20, 8))
sns.boxplot( data = dataframe, x="week", y="N_cases" )
ax.set_title( "Numero_casos por semana" )
```

Siendo la salida del código el siguiente gráfico:

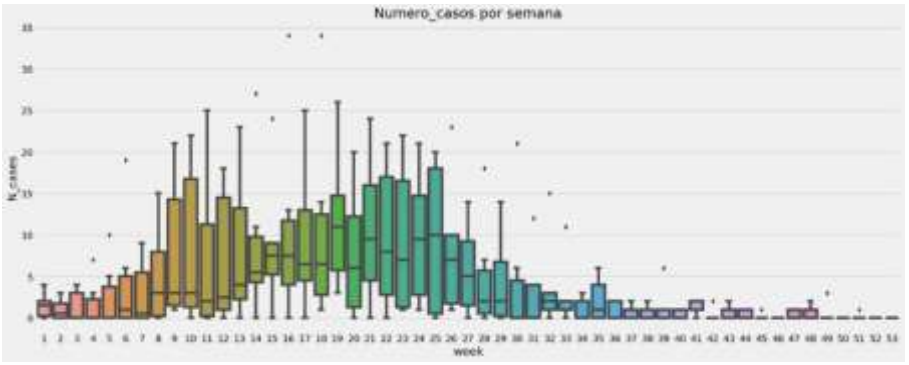


Figura 3
Casos de dengue en semanas ISO.

Si bien se identificó algunas particularidades en algunas fechas, se consideró mantenerlas.

Tuvimos que observar cómo se correlaciona las características a utilizar en relación al número de casos (N_cases), que nos permite descubrir las características que guardan relación a lo que se pretende predecir, usamos la correlación de Pearson para verificarlo.

```
plt.figure(figsize=(12, 6))
sns.heatmap(dataframe.corr(), cmap='coolwarm', annot=True,
center=0)
plt.show()
```

La salida del código es lo siguiente:

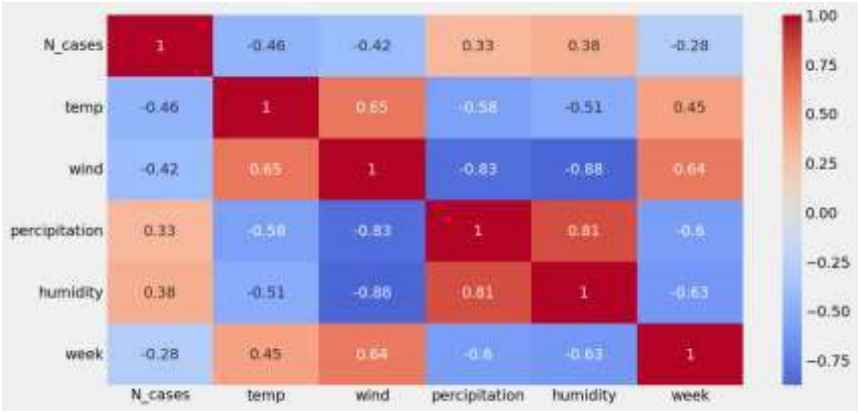


Figura 4
Características que conforman la data histórica.

Si bien el número de semana (“week”) no guarda mucha relación, por el objeto de estudio no podemos descartarlo, por ser la parte esencial para catalogarlo como serie temporal, las características que tuvimos en cuenta como candidatos a mantener sería “week”, “percipitation” y “humidity”, considerar que “N_cases” es el objetivo a predecir.

La partición de la data la elaboramos considerando que la problemática es de tipo serie temporal y que deseamos predecir hasta con 14 semanas de anticipación, siendo estas últimas las que empleamos para el test. Hicimos uso de la librería sklearn, particularmente la función train_test_split, que permite dividir el conjunto en entrenamiento y prueba, de manera que se ajuste a las condiciones mencionadas.

```
from sklearn.model_selection import train_test_split

X, y = dataframe[["week", "temp", "wind", "percipitation",
"humidity"]].values, dataframe['N_cases'].values
df_train, df_test = train_test_split(dataframe, test_size=14,
shuffle=False)

X_train, y_train = df_train[["week", "temp", "wind",
"percipitation", "humidity"]].values, df_train['N_cases'].values.T

X_test, y_test = df_test[["week", "temp", "wind", "percipitation",
"humidity"]].values, df_test['N_cases'].values.T
```

Así mismo, se reorganizó los subconjuntos de datos de entrenamiento y prueba para que funcionen correctamente al aplicarse el algoritmo XGBoost.

c) Procesamiento:

- **Determinación de parámetros.**

Para la configuración de los parámetros, se requirió importar el algoritmo XGBoost, y posteriormente emplear la clase “XGBRegressor”, para configurar los parámetros del modelo a construir, para este caso siendo:

- **n_estimators**: indica la cantidad de árboles de decisión a emplearse en el modelo, los árboles se afianzan para generar uno más potente.
- **max_depth**: permite establecer la profundidad máxima que cada árbol puede optar, es importante establecer la profundidad adecuada para prevenir el sobreajustamiento.
- **learning_rate**: está a cargo de controlar el aporte de cada árbol al modelo final, si el valor de este es bajo, generalizará mejor, sin embargo, afecta el rendimiento en hardware.
- **early_stopping_rounds**: ayuda en el control del entrenamiento del modelo, deteniendo anticipadamente a este de no existir alguna mejora, asume que el modelo ha convergido y no se debe seguir entrenando.
- **num_parallel_tree**: de acuerdo a las iteraciones de refuerzo, gestiona la cantidad de árboles paralelos construidos, contribuye en generalizar mejor el modelo.

De acuerdo a la casuística, vimos conveniente tener que elegir los hiperparámetros de “n_estimators” y “learning_rate”, que corresponde al número de árboles a emplear a lo largo del entrenamiento y el coeficiente de aprendizaje, con el propósito de mejorar el desempeño del modelo, para ello, planteamos diferentes valores que pueden asumir los parámetros en mención de la siguiente manera.

```
coeficiente_aprendizaje = [0.1, 0.05, 0.01, 0.005, 0.0001]
numero_estimaciones = [10, 100, 1000, 10000, 100000]
data_train = []
data_test = []
```

Definimos dos vectores para el número de estimaciones y el coeficiente de aprendizaje, y dos matrices vacías, que nos permitió recolectar los datos de los hiperparámetros con los resultados del entrenamiento del modelo.

Durante el entrenamiento del modelo también se realizó la combinación de los hiperparámetros, esto nos permitió entender como funcionaban en diferentes escenarios, con el siguiente código.

```
import xgboost as xgb

for i in range(len(coeficiente_aprendizaje)):
    for j in range(len(numero_estimaciones)):
        a = coeficiente_aprendizaje[i]
        b = numero_estimaciones[j]

        model = xgb.XGBRegressor(n_estimators=b,
                        max_depth=7,
                        learning_rate=a,
                        early_stopping_rounds=10,
                        num_parallel_tree=4)
        model.fit(X_train, y_train,
                        eval_set=[(X_train, y_train),
                        (X_test, y_test)], verbose=100)

        rmse_train = mean_squared_error(y_train,
model.predict(X_train), squared=False)
        rmse_test = mean_squared_error(y_test,
model.predict(X_test), squared=False)

        data_train.append([a, b, rmse_train])
        data_test.append([a, b, rmse_test])
```

Importamos el algoritmo XGBoost, empleamos dos bucles para recorrer los vectores de los hiperparámetros de coeficiente de aprendizaje y número de estimaciones respectivamente, por la casuística del problema empleamos la regresión del algoritmo, y los hiperparámetros como “max_depth”, “early_stopping_round” y “num_parallel_tree”, tuvieron los valores estáticos de 7, 10 y 4, mientras que

“n_estimators” y “learning_rate” será dinámico en relación a los vectores definidos previamente, esto es una configuración de la regresión de XGBoost, posteriormente entrenamos el modelo empleando la data segmentada y que a su vez permita visualizar la métrica RMSE tanto para datos de entrenamiento y test, además de ver la métrica de entrenamiento y test cada 100 árboles con el parámetro “verbose”.

Terminando el entrenamiento del modelo, obtenemos el RMSE en entrenamiento y prueba con la partición del objetivo a predecir en este caso “N_cases”, prosiguiendo, añadimos a los arrays vacíos, para almacenar la data de la métrica RMSE en función a los hiperparámetros “n_estimators” y “learning_rate”.

Hicimos uso de la siguiente variable para recuperar los datos de entrenamiento con pandas:

```
df_train = pd.DataFrame(data_train,
        columns=[       'CoeficienteAprendizaje',
                        'NumeroEstimadores',
                        'RMSE_Train'])
```

La variable contenía los siguientes datos en relación a los hiperparámetros y el RMSE del modelo con data de entrenamiento:

Tabla 5

Resultado de los hiperparámetros en data de entrenamiento

Coeficiente de aprendizaje	Númerode Estimadores	RMSE entrenamiento
0.1000	10	4.307547
0.1000	100	0.438328
0.1000	1000	0.438328
0.1000	10000	0.438328
0.1000	100000	0.438328
0.0500	10	5.873850
0.0500	100	2.172533
0.0500	1000	2.172533
0.0500	10000	2.172533
0.0500	100000	2.172533
0.0100	10	7.441407
0.0100	100	4.527441
0.0100	1000	2.654406
0.0100	10000	2.654406
0.0100	100000	2.654406
0.0050	10	7.671138
0.0050	100	5.920266
0.0050	1000	2.342092

0.0050	10000	2.342092
0.0050	100000	2.342092
0.0001	10	7.909005
0.0001	100	7.864377
0.0001	1000	7.443414
0.0001	10000	4.424032
0.0001	100000	2.771626

Hicimos uso de la siguiente variable para recuperar los datos de entrenamiento con pandas:

```
df_test = pd.DataFrame(data_test,
        columns=[       'CoeficienteAprendizaje',
                        'NumeroEstimadores',
                        'RMSE_Test'])
```

La variable contenía los siguientes datos en relación a los hiperparámetros y el RMSE del modelo con data de prueba:

Tabla 6

Resultado de los hiperparámetros en data de prueba

Coeficiente de aprendizaje	**Númerode Estimadores**	**RMSE prueba**
0.1000	10	9.926970
0.1000	100	9.380412
0.1000	1000	9.380412
0.1000	10000	9.380412
0.1000	100000	9.380412
0.0500	10	10.625387
0.0500	100	9.224757
0.0500	1000	9.224757
0.0500	10000	9.224757
0.0500	100000	9.224757
0.0100	10	11.171102
0.0100	100	9.764701
0.0100	1000	9.393394
0.0100	10000	9.393394
0.0100	100000	9.393394
0.0050	10	11.253290
0.0050	100	10.548876
0.0050	1000	9.223042

0.0050	10000	9.223042
0.0050	100000	9.223042
0.0001	10	11.341776
0.0001	100	11.324993
0.0001	1000	11.171800
0.0001	10000	9.755798
0.0001	100000	9.405089

- **Entrenamiento del modelo.**

Los hiperparámetros en relación a “n_estimators”, “learning” y el valor de la métrica del RMSE tanto entrenamiento y prueba, además, considerando el margen para el resto de modelos, los hiperparámetros a emplear para posteriores modelos son “n_estimators” con 10000 y “learning_rate” con 0.005.

Entrenamos el modelo con los hiperparámetros elegidos:

```
model = xgb.XGBRegressor(n_estimators=10000,
                                 max_depth=7,
                                 learning_rate=0.005,
                                 early_stopping_rounds=10,
                                 num_parallel_tree=4)
model.fit(X_train, y_train,
                eval_set=[(X_train, y_train),
                (X_test, y_test)],
                verbose=100)
```

- **Validación de características.**

Verificamos si las características previamente seleccionadas son importantes en el entrenamiento del modelo:

```
importancias =
model.get_booster().get_score(importance_type='weight')
from xgboost import plot_importance
nombre_caracteristicas = ["week", "temp", "wind", "percipitation",
"humidity"]
i = 0
for caracteristica, importancia in importancias.items():
    print('Característica: %s, Score: %.5f' % ( caracteristica +
":" + nombre_caracteristicas[i], importancia))
    i = i + 1
plt.figure(figsize=(12, 8))
plot_importance(model, importance_type='weight',
max_num_features=12)
plt.show()
```

La salida del código es lo siguiente:

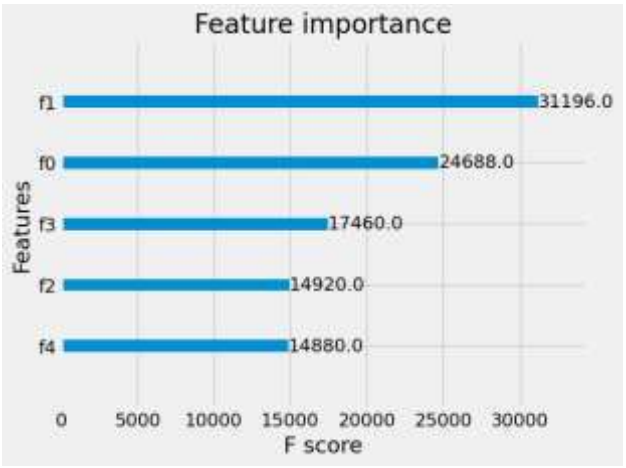


Figura 5
Validación de características para el modelo AR.

Nos permitió entender que la característica “week” (semana), es importante en el entrenamiento del modelo, “temp”, “wind”, “percipitation” y “humidity”, tienen valores a considerar, sin embargo, debido a la previa correlación de Pearson con el objetivo a predecir reflejado en el gráfico de calor, decidimos emplear las características “week”, “percipitation” y “humidity” para posteriores modelos.

3.3.2 Objetivo específico 2

Utilizar datos de fuentes alternativas fiables para afianzar la data histórica existente y garantizar mayor precisión de la predicción.

Las actividades del objetivo se explican según el diagrama:

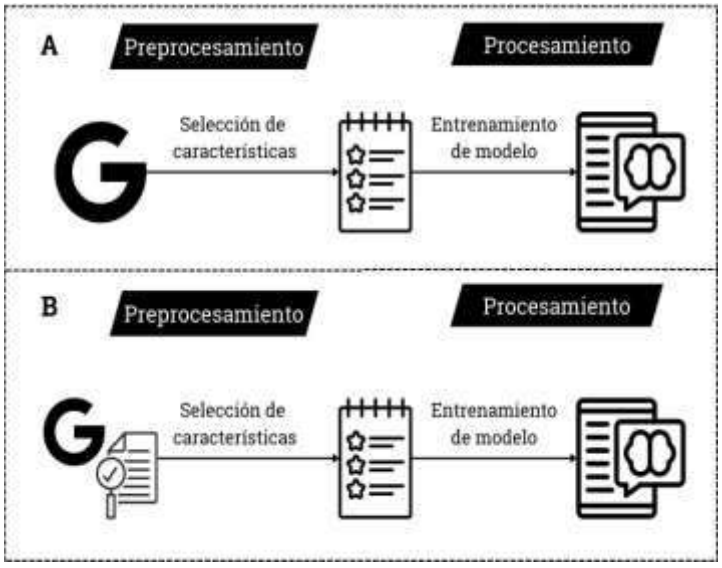


Figura 6
Diagrama de pasos del objetivo específico 2.

A) Modelo entrenado con las "queries" de Google Trends.

a) Preprocesamiento:

- **Selección de características.**

Decidimos emplear las palabras buscadas por Google Trends en relación a cada hoja del set de datos correspondiente a cada ciudad, para la creación de este modelo, el punto de partida fue una ciudad de ejemplo, "Maranguape" para este caso, empleamos las mismas librerías que el modelo anterior.

Seleccionamos los datos en relación a la ciudad y columnas a emplear, en este caso las queries obtenidas de Google Trends.

```
dataframe = pd.read_excel(r'Dengue_selected_cities.xlsx',
sheet_name = "Maranguape", usecols = ["date", "N_cases","dengue",
"a dengue", "sintomas da dengue", "sintomas dengue", "sobre a
dengue", "sintomas de dengue", "Mosquito da dengue", "Mosquito
Dengue", "dengue hemorragica", "mosquito", "mosquitos" ]
)
```

Establecimos la fecha como índice para posteriormente, seleccionar datos solo a partir del 2011 en adelante.

```
dataframe = dataframe.set_index("date")
dataframe = dataframe.loc[ dataframe.index > "2011-01-01"]
```

Tuvimos que observar cómo se correlaciona las características a utilizar en relación al número de casos (N_cases), que nos permite descubrir las características que guardan relación a lo que se pretende predecir, usamos la correlación de Pearson para verificarlo.

```
plt.figure(figsize=(12, 6))
sns.heatmap(dataframe.corr(), cmap='coolwarm', annot=True,
center=0)
plt.show()
```

La salida del código es lo siguiente:

Figura 7
Características que conforman la data de Google Trends.

Comprobamos que las “queries” de Google Trends tienen una correlación positiva en función de los “N_cases”, por ello, para esta investigación consideramos incluir en todos los entrenamientos que competan a estas características en el resto de modelos que las empleen.

La partición de la data la elaboramos considerando que la problemática es de tipo serie temporal y que deseamos predecir hasta con 14 semanas de anticipación.

```
df_train, df_test = train_test_split(dataframe, test_size=14,
shuffle=False)
X_train, y_train = df_train[["dengue", "a dengue", "sintomas da
dengue", "sintomas dengue", "sobre a dengue",
                        "sintomas de dengue", "Mosquito da dengue",
"Mosquito Dengue", "dengue hemorragica", "mosquito",
                        "mosquitos"]].values,
df_train['N_cases'].values.T

X_test, y_test = df_test[["dengue", "a dengue", "sintomas da
dengue", "sintomas dengue", "sobre a dengue",
                        "sintomas de dengue", "Mosquito da dengue",
"Mosquito Dengue", "dengue hemorragica", "mosquito",
                        "mosquitos"]].values,
df_test['N_cases'].values.T
```

Así mismo, se reorganizó los subconjuntos de datos de entrenamiento y prueba para que funcionen correctamente al aplicarse el algoritmo XGBoost.

b) Procesamiento:

- **Entrenamiento del modelo.**

Durante el entrenamiento del modelo empleamos la combinación de hiperparámetros del modelo anterior.

```
model = xgb.XGBRegressor( n_estimators = 10000,
                          max_depth = 7,
                          learning_rate = 0.005,
                          early_stopping_rounds = 10,
                          num_parallel_tree = 4 )
model.fit(      X_train, y_train,
                eval_set = [ (X_train, y_train),(X_test, y_test) ],
                verbose = 20)
```

B) Modelo entrenado con la data histórica y las "queries" de Google Trends.

a) Preprocesamiento:

- **Selección de características.**

Seleccionamos los datos en relación a la ciudad en este caso "Maranguape" y columnas a emplear, en este caso la data histórica y las queries obtenidas de Google Trends considerando las características decididas a emplear.

```
dataframe = pd.read_excel(r'Dengue_selected_cities.xlsx',
sheet_name = "Maranguape", usecols = ["date", "N_cases","dengue",
"a dengue", "sintomas da dengue", "sintomas dengue", "sobre a
dengue", "sintomas de dengue", "Mosquito da dengue", "Mosquito
Dengue", "dengue hemorragica", "mosquito", "mosquitos",
"percipitation", "humidity" ])
```

Establecimos la fecha como índice para posteriormente, seleccionar datos solo a partir del 2011 en adelante.

```
dataframe = dataframe.set_index("date")
dataframe = dataframe.loc[ dataframe.index > "2011-01-01"]
```

Tuvimos que observar cómo se correlaciona las características a utilizar en relación al número de casos (N_cases), que nos permite descubrir las características que guardan relación a lo que se pretende predecir, usamos la correlación de Pearson para verificarlo.

```
plt.figure(figsize=(12, 6))
sns.heatmap(dataframe.corr(), cmap='coolwarm', annot=True,
center=0)
plt.show()
```

La salida del código es lo siguiente:

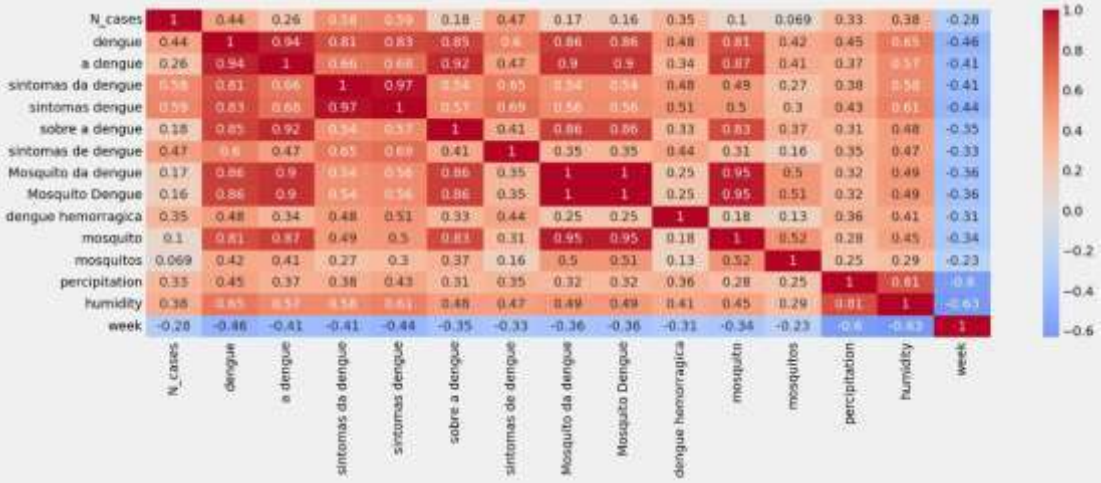


Figura 8
Características que conforman la data histórica y de Google Trends.

Comprobamos que las características tienen una correlación positiva a excepción de "week", que anteriormente brindamos los motivos de su preservación.

La partición de la data la elaboramos considerando que la problemática es de tipo serie temporal y que deseamos predecir hasta con 14 semanas de anticipación.

```
df_train, df_test = train_test_split(dataframe, test_size=14,
shuffle=False)
X_train, y_train = df_train[['week', "dengue", "a dengue",
"sintomas da dengue", "sintomas dengue", "sobre a dengue",
"sintomas de dengue", "Mosquito da dengue", "Mosquito Dengue",
"dengue hemorragica", "mosquito", "mosquitos", "percipitation",
"humidity"]].values, df_train['N_cases'].values.T

X_test, y_test = df_test[['week', "dengue", "a dengue",
"sintomas da dengue", "sintomas dengue", "sobre a dengue",
"sintomas de dengue", "Mosquito da dengue", "Mosquito Dengue",
"dengue hemorragica", "mosquito", "mosquitos", "percipitation",
"humidity"]].values, df_test['N_cases'].values.T
```

Así mismo, se reorganizó los subconjuntos de datos de entrenamiento y prueba para que funcionen correctamente al aplicarse el algoritmo XGBoost.

b) Procesamiento:

- **Entrenamiento del modelo.**

Durante el entrenamiento del modelo empleamos la combinación de hiperparámetros del modelo anterior.

```
model = xgb.XGBRegressor( n_estimators = 10000,
                          max_depth = 7,
                          learning_rate = 0.005,
                          early_stopping_rounds = 10,
                          num_parallel_tree = 4 )
model.fit(      X_train, y_train,
                eval_set = [ (X_train, y_train),(X_test, y_test) ],
                verbose = 20)
```

3.3.3 Objetivo específico 3

Construir una herramienta tecnológica basado en dashboard para mostrar los indicadores de los modelos en la predicción de la incidencia del dengue.

Las actividades del objetivo se explican según el diagrama:

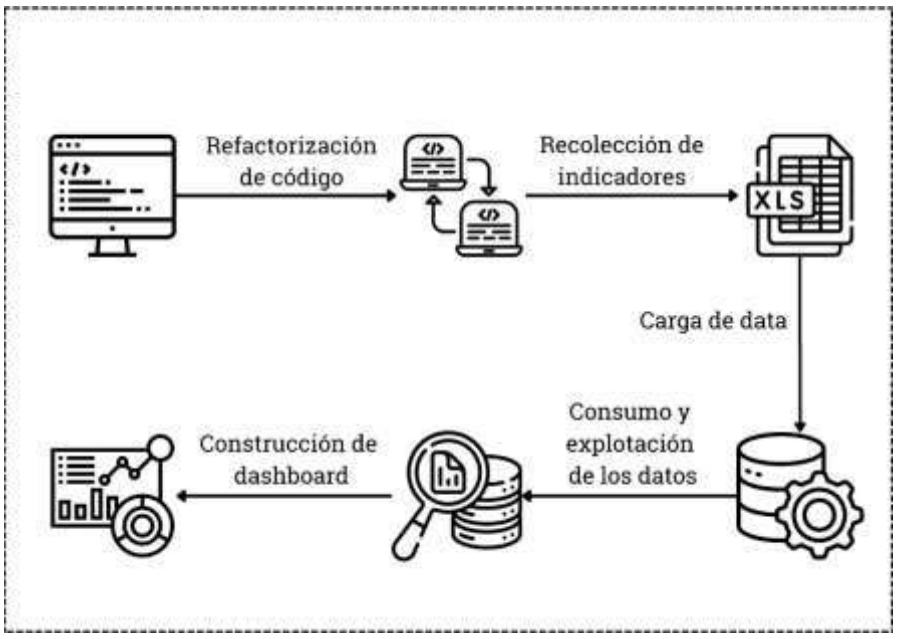


Figura 9
Diagrama de pasos del objetivo específico 3.

a) Refactorización del código.

Para la refactorización de código tomamos en cuenta los datos recopilados en relación a las características e hiperparámetros establecidos en los anteriores objetivos, creamos un array con el nombre de las hojas del dataset en relación a las ciudades y correspondiente a cada modelo para entrenar los modelos de manera recursiva respecto a determinada ciudad.

b) Recolección de métricas.

La librería que en esta casuística utilizamos fue la de openpyxl, que nos permitió manipular archivos Excel, incrustamos el código para que cada métrica sea obtenida correctamente y almacenar los datos de las métricas en relación al algoritmo, modelo,

predicción en datos de entrenamiento o prueba (“test”), nombre de la métrica y el respectivo valor, el código está disponible en el Anexo 2.

Los datos recopilados tienen la siguiente estructura en el archivo Excel:

Tabla 7

Ejemplo de data recopilada

ciudad	algoritmo	modelo	data	indicador	valor
Ji Parana	XGBoost	Modelo 1 - AR	Entrenamiento	RMSE	17.47963
Ji Parana	XGBoost	Modelo 1 - AR	Entrenamiento	R-RMSE	0.131426
Ji Parana	XGBoost	Modelo 1 - AR	Entrenamiento	R^2	0
Ji Parana	XGBoost	Modelo 1 - AR	Entrenamiento	C. Pearson	0.954937
Ji Parana	XGBoost	Modelo 1 - AR	Test	RMSE	0.501611
Ji Parana	XGBoost	Modelo 1 - AR	Test	R-RMSE	0.501611
Ji Parana	XGBoost	Modelo 1 - AR	Test	R^2	0
Ji Parana	XGBoost	Modelo 1 - AR	Test	C. Pearson	-0.00929

En la tabla es solo un ejemplo de las métricas obtenidas de solo una ciudad siendo para el ejemplo “Ji Parana”, es decir, que cada una de las 20 ciudades tienen la misma cantidad de información recopilada.

c) Carga de la data.

Usamos un servidor de base de datos de MySQL en un servidor privado de versión 10.3.39 - MariaDB - MariaDB Server, creamos la base de datos y la tabla necesaria, esta última tiene el siguiente esquema:

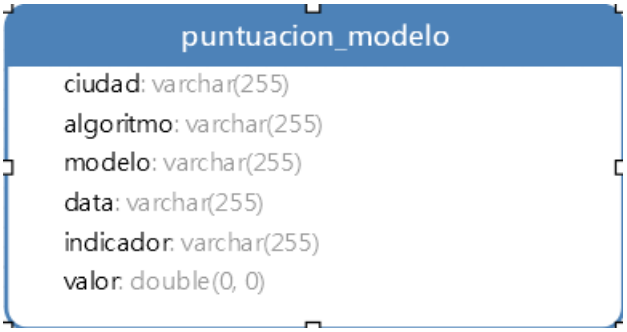


Figura 10
Tabla puntuacion_modelo.

La tabla tiene como nombre “puntuacion_modelo”, con los siguientes caracteres: “ciudad” es un varchar con la capacidad de hasta 255 caracteres, “algoritmo” es un varchar con la capacidad de hasta 255 caracteres, “modelo” es un varchar con la capacidad de hasta 255 caracteres, “data” es un varchar con la capacidad de hasta

255 caracteres, “indicador” es un varchar con la capacidad de hasta 255 caracteres, y “valor” es un double por la existencia de decimales.

Empleamos el asistente exportación de Navicat 12.0.23 de 64 bits, para cargar los datos desde el archivo Excel:

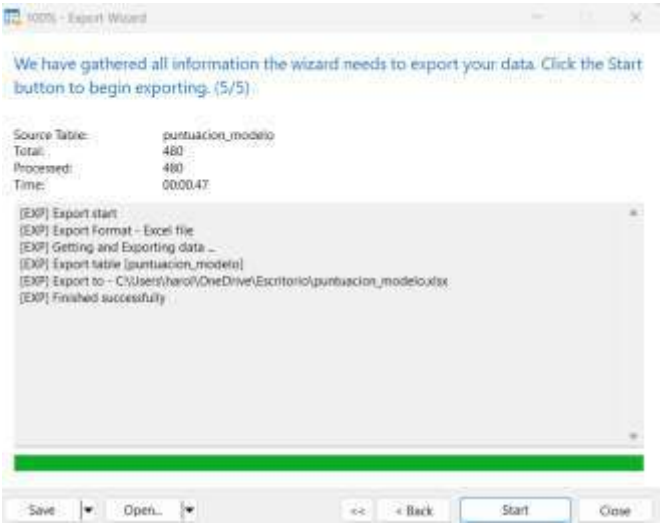


Figura 11
Exportación de datos a MySQL.
Exportamos un total de 480 datos.

Una muestra de los datos en la base de datos MySQL:

ciudad	algoritmo	modelo	data	indicador	valor
Parana	XGBoost	Modelo 1 - AR	Entrenamiento	RMSE	17.479633
Parana	XGBoost	Modelo 1 - AR	Entrenamiento	R-RMSE	0.131426
Parana	XGBoost	Modelo 1 - AR	Entrenamiento	R^2	0
Parana	XGBoost	Modelo 1 - AR	Entrenamiento	C. Pearson	0.954937
Parana	XGBoost	Modelo 1 - AR	Test	RMSE	0.501611
Parana	XGBoost	Modelo 1 - AR	Test	R-RMSE	0.501611
Parana	XGBoost	Modelo 1 - AR	Test	R^2	0
Parana	XGBoost	Modelo 1 - AR	Test	C. Pearson	-0.009289
Parana	XGBoost	Modelo 2 - GT	Entrenamiento	RMSE	17.480763
Parana	XGBoost	Modelo 2 - GT	Entrenamiento	R-RMSE	0.131434
Parana	XGBoost	Modelo 2 - GT	Entrenamiento	R^2	0
Parana	XGBoost	Modelo 2 - GT	Entrenamiento	C. Pearson	0.962655
Parana	XGBoost	Modelo 2 - GT	Test	RMSE	0.503823
Parana	XGBoost	Modelo 2 - GT	Test	R-RMSE	0.503823
Parana	XGBoost	Modelo 2 - GT	Test	R^2	0
Parana	XGBoost	Modelo 2 - GT	Test	C. Pearson	-0.41442
Parana	XGBoost	Modelo 3 - AR+GT	Entrenamiento	RMSE	17.477466
Parana	XGBoost	Modelo 3 - AR+GT	Entrenamiento	R-RMSE	0.13141
Parana	XGBoost	Modelo 3 - AR+GT	Entrenamiento	R^2	0
Parana	XGBoost	Modelo 3 - AR+GT	Entrenamiento	C. Pearson	0.930887
Parana	XGBoost	Modelo 3 - AR+GT	Test	RMSE	0.504625
Parana	XGBoost	Modelo 3 - AR+GT	Test	R-RMSE	0.504625
Parana	XGBoost	Modelo 3 - AR+GT	Test	R^2	0
Parana	XGBoost	Modelo 3 - AR+GT	Test	C. Pearson	-0.359914

Figura 12

Vista previa de datos en MySQL.

d) Consumo y explotación de los datos.

Empleamos el software de Power Bi versión 2.109.782.0 de 64 bits para el desarrollo de este apartado. Conectamos con la base de datos y cargamos los datos a la herramienta:

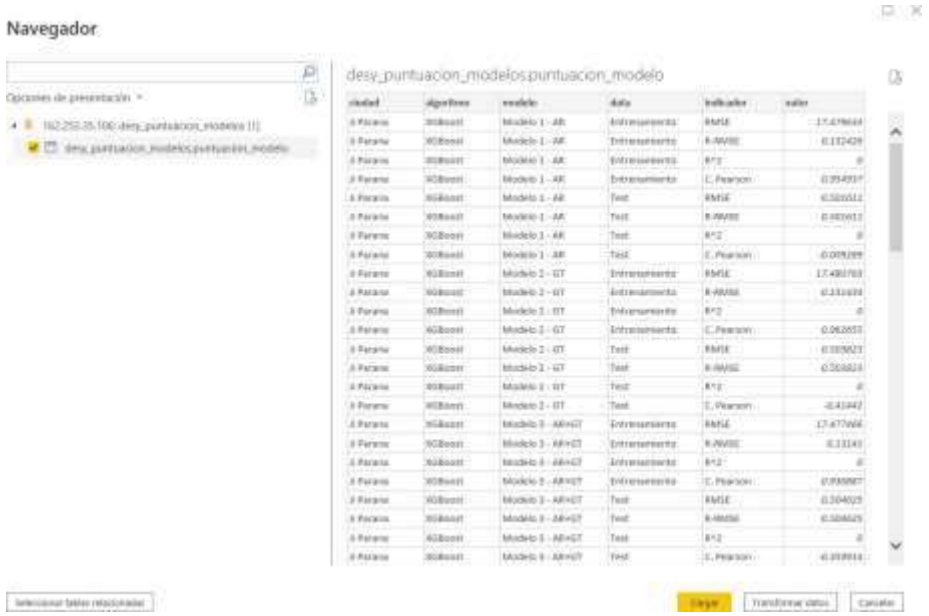


Figura 13
Vista previa de datos en Power BI.

Para tener un mejor enfoque, consideramos transformar las métricas que pertenecen al campo indicador (RMSE, R-RMSE, R^2 y Correlación de Pearson), en columnas, esto para tener una comparación basada en varias métricas a la vez.

CAPÍTULO IV
RESULTADOS Y DISCUSIÓN

4.1 Resultado específico 1

Siendo nuestro propósito el obtener las métricas de nuestro modelo empleando los datos históricos del dataset, que ayuda en el contraste respecto a próximos modelos.

Empleamos las siguientes librerías:

```
from sklearn.metrics import mean_squared_error, r2_score
from scipy.stats import pearsonr
```

Obtención de métricas de predicción del modelo en data de entrenamiento:

```
predictions = model.predict(X_train)
mse = mean_squared_error(y_train, predictions)
rmse = np.sqrt(mse)
print("RMSE:", rmse)
r2 = r2_score(y_train, predictions)
print("R2:", r2)
print("R-RMSE:", relative_root_mean_squared_error(rmse, y_train))
print('Correlación de Pearson:', pearsonr(y_train,
predictions)[0])
```

Obtuvimos mejoras significativas reflejadas en las métricas en el modelo planteado en esta investigación en relación a la ciudad “Eunapolis” respecto al de (Koplewitz et al., 2022), quien tiene las siguientes métricas:

Tabla 8

Versus de métricas recopiladas del modelo con data histórica de entrenamiento (Eunapolis)

Métrica	**Nuestro modelo**	**Modelo de** (Koplewitz et al., 2022)
RMSE	2.3420	4.617
R2	0.8855	0.556
R-RMSE	0.0688	0.582
Correlación de Pearson	0.9774	0.752

Siendo la tendencia de la predicción la siguiente:

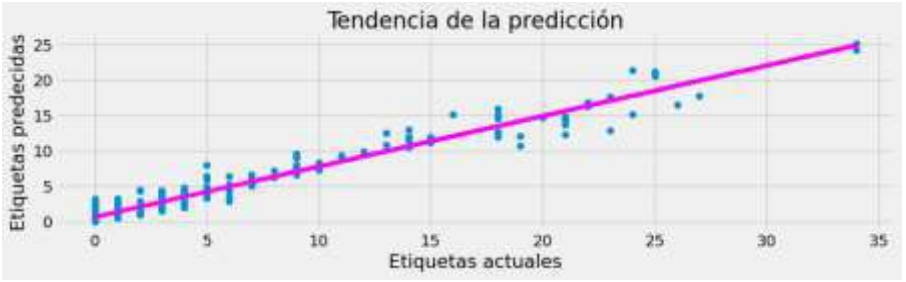


Figura 14

Tendencia de predicción del modelo con data histórica de entrenamiento.

Observamos que la tendencia de la predicción es relativamente buena, con una generalización a crecer, existiendo variaciones en relación a las etiquetas actuales y las predichas.

El ajuste del modelo es el siguiente:

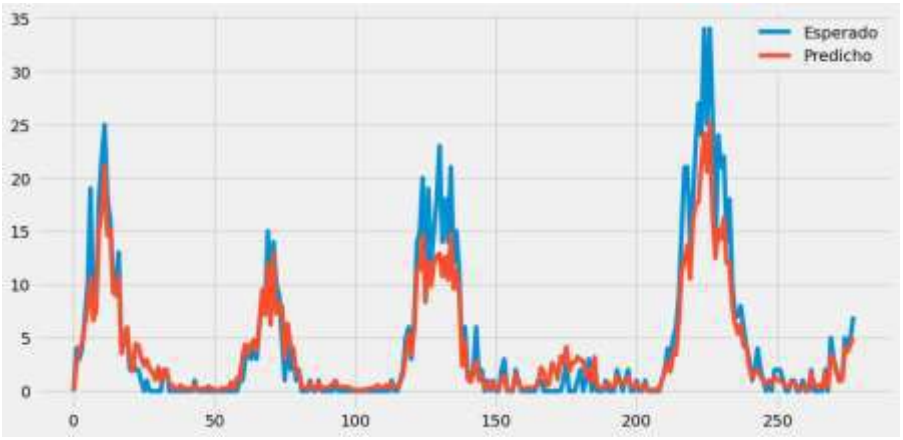


Figura 15

Ajuste del modelo con data histórica de entrenamiento.

El gráfico de color azul pertenece al número de casos del dataset o “N_cases” y gráfico de color anaranjado es la predicción del modelo, destaca en picos menores y no se ajusta adecuadamente en picos altos de incidencia de dengue.

Obtención de métricas de predicción del modelo en data de prueba:

```
predictions = model.predict(X_test)
predictions_test = predictions
def relative_root_mean_squared_error(rmse, y_test):
    den = np.max(y_test) - np.min(y_test)
    return (rmse/den)
mse = mean_squared_error(y_test, predictions)
rmse = np.sqrt(mse)
print("RMSE:", rmse)
r2 = r2_score(y_test, predictions)
print("R2:", r2)
print("R-RMSE:", relative_root_mean_squared_error(rmse, y_test))
print('Correlación de Pearson:', pearsonr(y_test, predictions)[0])
```

Tuvimos como resultado lo siguiente:

Tabla 9

Métricas recopiladas del modelo con data histórica de prueba (Eunapolis)

Métrica	Valor
RMSE	9.2230
R2	0.0
R-RMSE	0.4391
Correlación de Pearson	-0.0761

Siendo la tendencia de la predicción la siguiente:

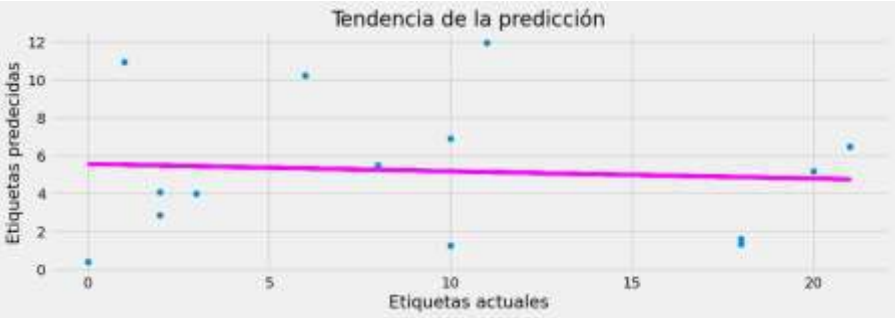


Figura 16

Tendencia de predicción del modelo con data histórica de prueba.

La tendencia de la predicción se orienta al decrecimiento, esto debe a la variabilidad de los datos objetivos (reales) de predicción y actuales en la data de prueba.

El modelo predice de la siguiente manera:

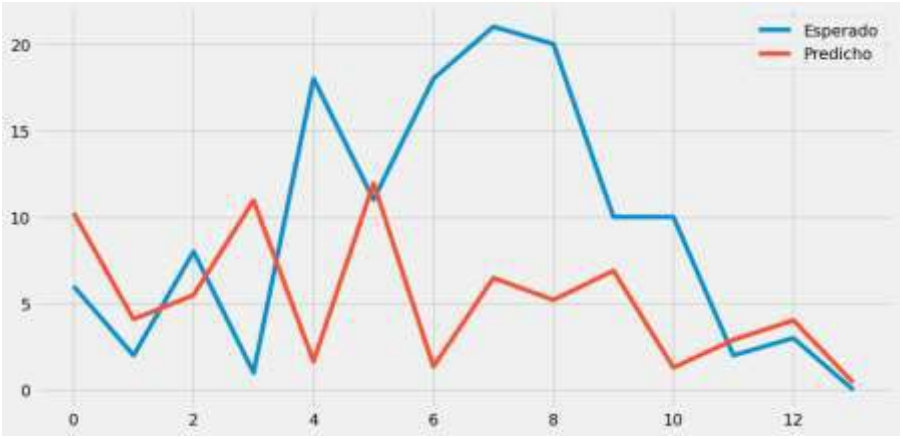


Figura 17

Ajuste del modelo con data histórica de prueba.

El gráfico de color azul corresponde a data del dataset y el gráfico de color anaranjado corresponde a las predicciones realizadas por el modelo, este último predice correctamente algunas incidencias de dengue, sin embargo, es importante reconocer los valores erráticos de los números de casos del dataset.

Los gráficos generales respecto a las incidencias reales utilizadas para la prueba y la predicción segmentado considerando las 14 semanas:

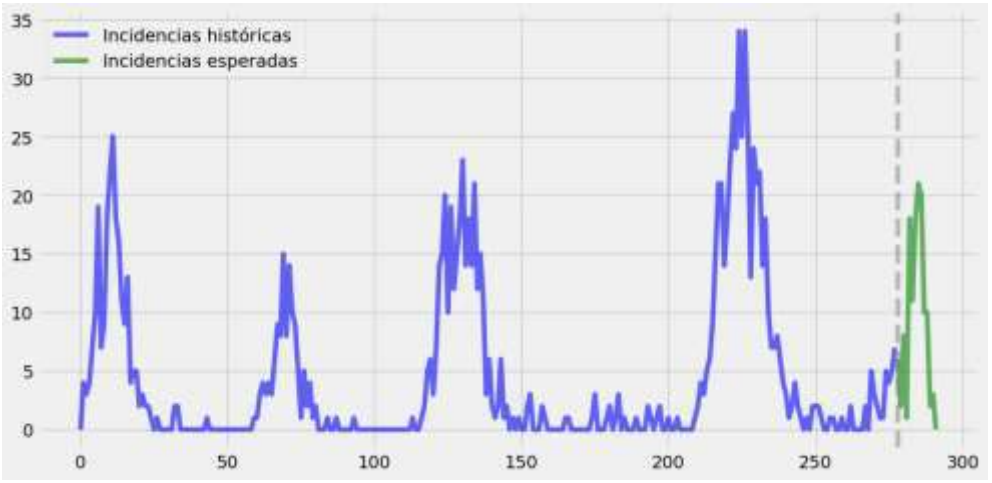


Figura 18

Predicciones esperadas.

Según este gráfico, las “Incidencias esperadas” dan a entender que estaba por ocurrir un pico de casos considerables que disminuyen posteriormente.

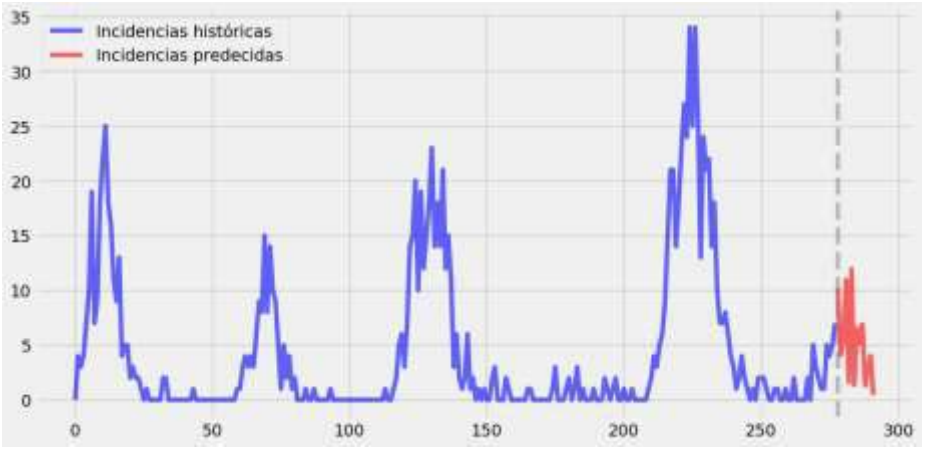


Figura 19

Predicciones realizadas con datos históricos.

Según las predicciones, el fenómeno del pico de incidencias no es capturado correctamente sin embargo el modelo sí predice adecuadamente la disminución en el periodo dado, esto puede darse debido a lo errático de los datos recopilados y/o fenómeno de incremento abrupto de incidencias de dengue.

El empleo de características para el entrenamiento de este modelo está conforme a investigaciones con similar casuística, como el de (Souza et al., 2022) quienes consideran que se debe emplear periodos semanales y datos climatológicos en esta problemática.

4.2 Resultado específico 2

a) Modelo entrenado con las "queries" de Google Trends.

Las métricas de predicción del modelo en data de entrenamiento respecto a este modelo, tenemos mejoras significativas reflejadas en las métricas en el modelo planteado en esta investigación en relación a la ciudad "Eunapolis" respecto al de (Koplewitz et al., 2022), quien tiene las siguientes métricas:

Tabla 10

Versus de métricas recopiladas del modelo con data de Google Trends de entrenamiento (Eunapolis)

Métrica	**Nuestro modelo**	**Modelo de** (Koplewitz et al., 2022)
RMSE	2.8542	6.751
R2	0.8300	0.051
R-RMSE	0.0839	0.851
Correlación de Pearson	0.9853	0.52

Siendo la tendencia de la predicción la siguiente:

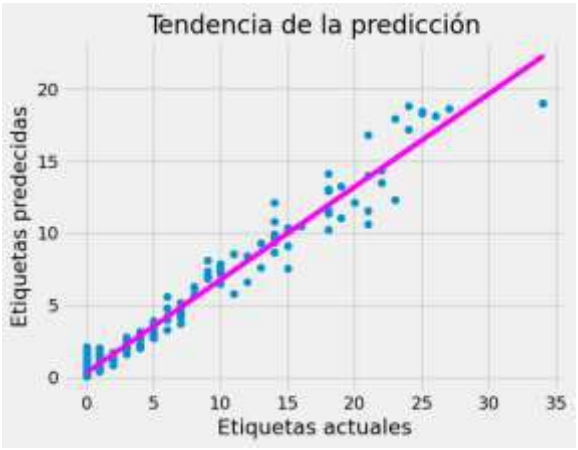


Figura 20

Tendencia de predicción del modelo con data de Google Trends de entrenamiento.

En una primera instancia, observamos que la tendencia de la predicción es relativamente buena, con una generalización a crecer, con variaciones en relación al objetivo y la predicción.

El modelo predice de la siguiente manera:

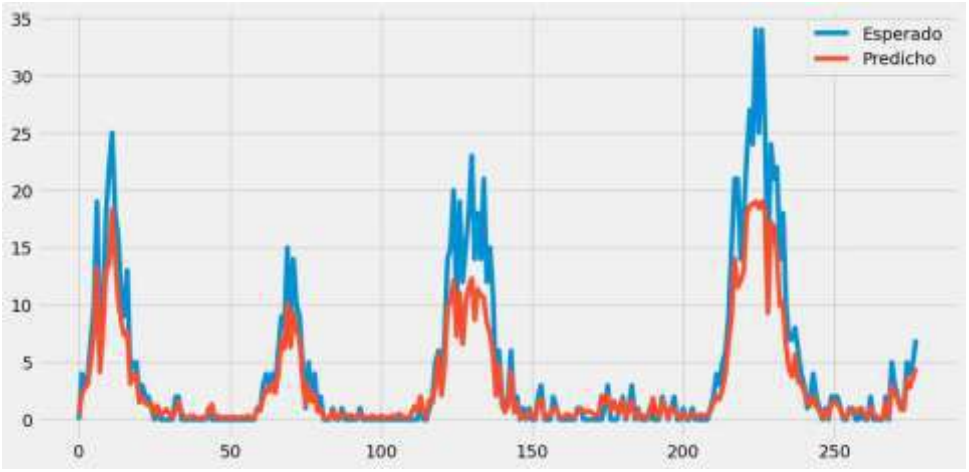


Figura 21

Ajuste del modelo con data de Google Trends de entrenamiento.

El gráfico de color azul pertenece al número de casos del dataset o “N_cases” y gráfico de color anaranjado es la predicción del modelo, destaca en picos menores y no se ajusta adecuadamente en picos altos de incidencia de dengue, el motivo de la verificación es para conocer si el modelo está sobreajustándose.

Obtención de métricas de predicción del modelo en data de prueba:

Tabla 11

Métricas recopiladas del modelo con data de Google Trends de prueba (Eunapolis)

Métrica	Valor
RMSE	9.3458
R2	0.0
R-RMSE	0.4450
Correlación de Pearson	0.0523

Siendo la tendencia de la predicción la siguiente:

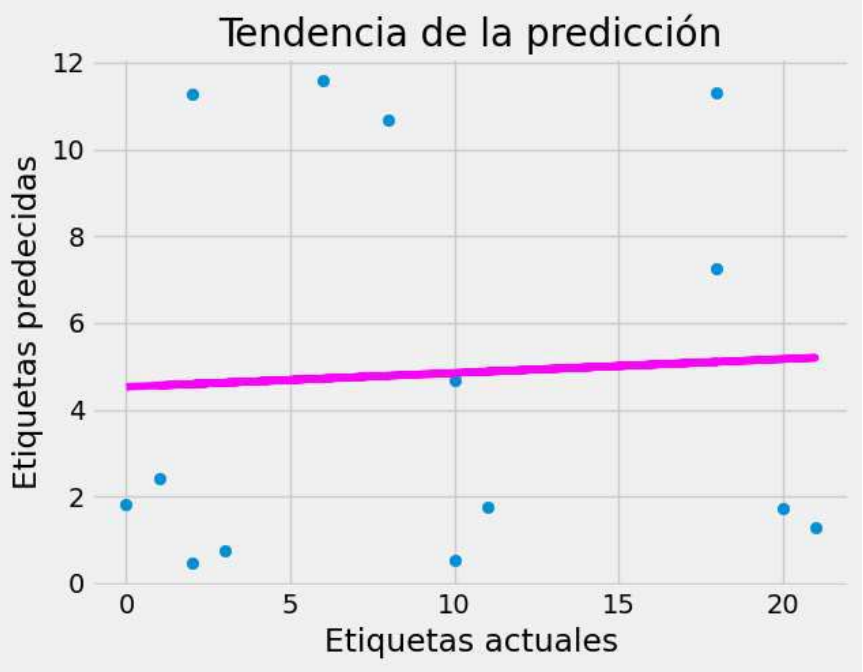


Figura 22

Tendencia de predicción del modelo con data de Google Trends de prueba.

La tendencia de la predicción trata de tender a crecimiento, como lo explica en el apartado anterior de predicción del modelo con data de entrenamiento, sin embargo, hay una importante variación entre los datos de predicción y actuales en la data de prueba.

El modelo predice de la siguiente manera:

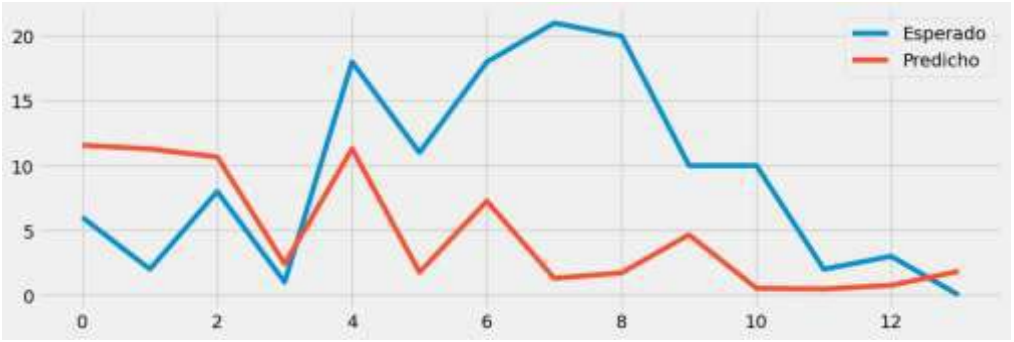


Figura 23

Ajuste del modelo con data de Google Trends de prueba.

El gráfico de color azul corresponde a data del dataset y el gráfico de color anaranjado corresponde a las predicciones realizadas por el modelo, este último se acerca mucho a algunas incidencias de dengue.

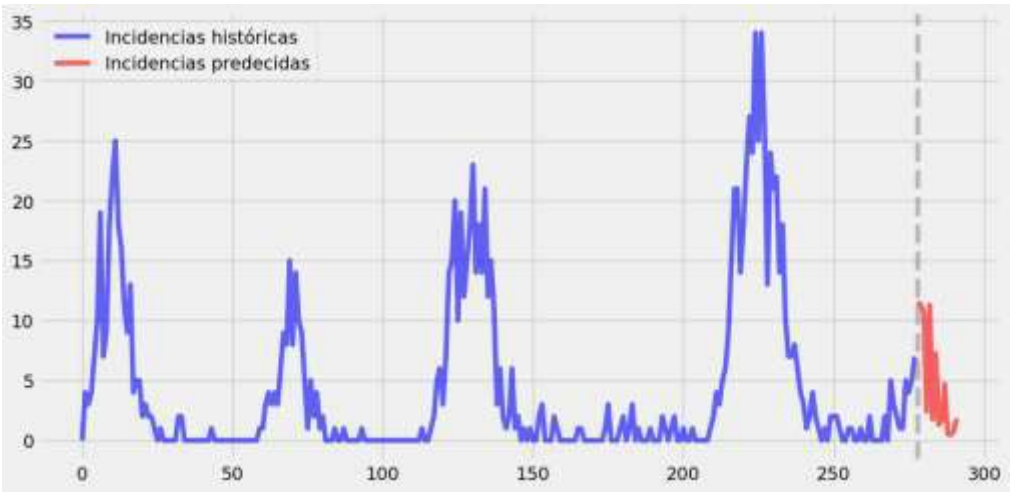


Figura 24

Predicciones realizadas con data de Google Trends.

Según las predicciones, el fenómeno del pico de incidencias no es capturado correctamente, sin embargo; el modelo sí predice adecuadamente la disminución en el periodo dado.

b) Modelo entrenado con la data histórica y las “queries” de Google Trends.

Las métricas de predicción del modelo en data de entrenamiento respecto a este modelo, tenemos mejoras significativas reflejadas en las métricas en el modelo planteado en esta investigación en relación a la ciudad “Eunapolis” respecto al de (Koplewitz et al., 2022), quien tiene las siguientes métricas:

Tabla 12

Versus de métricas recopiladas del modelo con data histórica y de Google Trends de entrenamiento (Eunapolis)

Métrica	**Nuestro modelo**	**Modelo de** (Koplewitz et al., 2022)
RMSE	1.1213	4.718
R2	0.9737	0.536
R-RMSE	0.0329	0.595
Correlación de Pearson	0.9971	0.744

Siendo la tendencia de la predicción la siguiente:

Figura 25

Tendencia de predicción del modelo con data histórica y de Google Trends de entrenamiento.

En una primera instancia, observamos que la tendencia de la predicción es relativamente buena, con una generalización a crecer, con pequeñas variaciones respecto a la recta.

El modelo predice de la siguiente manera:

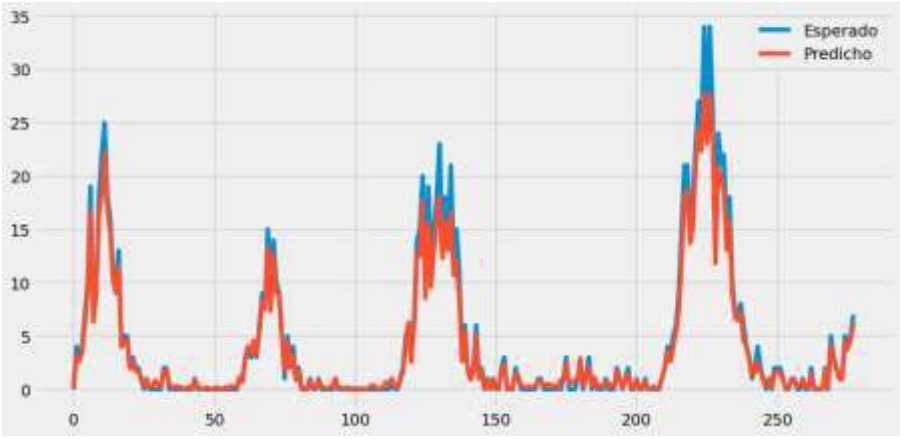


Figura 26
Ajuste del modelo con data histórica y de Google Trends de entrenamiento.

El gráfico de color azul pertenece al número de casos del dataset o “N_cases” y gráfico de color anaranjado es la predicción del modelo, destaca en picos menores y picos altos de incidencia de dengue, el motivo de la verificación es para conocer si el modelo está sobreajustándose.

Obtención de métricas de predicción del modelo en data de prueba:

Tabla 13

Métricas recopiladas del modelo con data histórica y de Google Trends de prueba (Eunapolis)

Métrica	**Valor**
RMSE	8.2301
R2	0.0
R-RMSE	0.3919
Correlación de Pearson	0.1817

Siendo la tendencia de la predicción la siguiente:

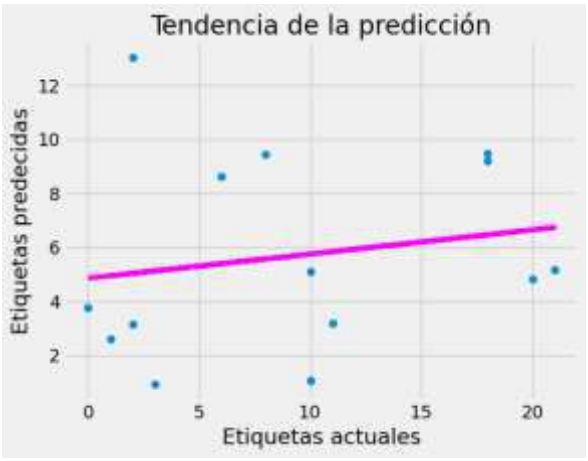


Figura 27

Tendencia de predicción del modelo con data histórica y de Google Trends de prueba.

La tendencia de la predicción trata de tender a crecimiento, como lo explica en el apartado anterior de predicción del modelo con data de entrenamiento, sin embargo, hay una importante variación entre los datos de predicción y actuales en la data de prueba.

El modelo predice de la siguiente manera:

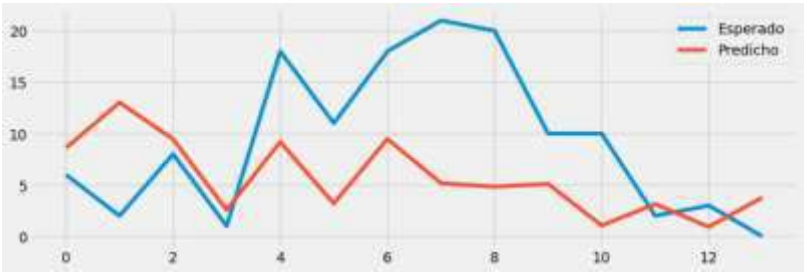


Figura 28

Ajuste del modelo con data histórica y de Google Trends de prueba.

El gráfico de color azul corresponde a data del dataset y el gráfico de color anaranjado corresponde a las predicciones realizadas por el modelo, este último se acerca mucho a algunas incidencias de dengue.

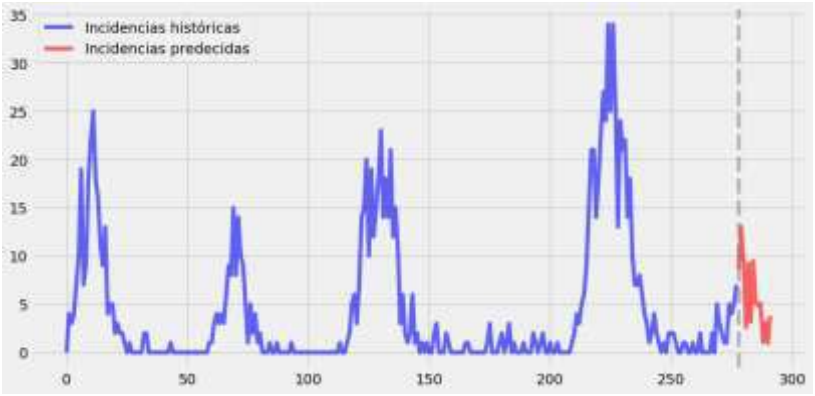


Figura 29

Predicciones realizadas con data histórica y de Google Trends.

Según las predicciones, el fenómeno del pico de incidencias no es capturado correctamente sin embargo el modelo sí predice adecuadamente la disminución en el periodo dado, esto puede darse debido a lo errático de los datos recopilados y/o fenómeno de incremento abrupto de incidencias de dengue incidencias de dengue.

(Aiken et al., 2020) menciona que los modelos que emplean datos de fuentes alternativas como Google Trends, realizan estimaciones razonables dentro del periodo de actividad de la enfermedad, con un modelo basado en árboles tienen como RMSE 17,63 y con R-RMSE 0,56 en datos de prueba, y nuestro modelo en esta casuística tiene de RMSE 9.3458 y R-RMSE 0,4450.

4.3 Resultado específico 3

Construimos un dashboard con los datos transformados, para nosotros era importante poder vislumbrar las 4 métricas definidas en su conjunto ya que permite afianzar la capacidad de predicción de determinado modelo.

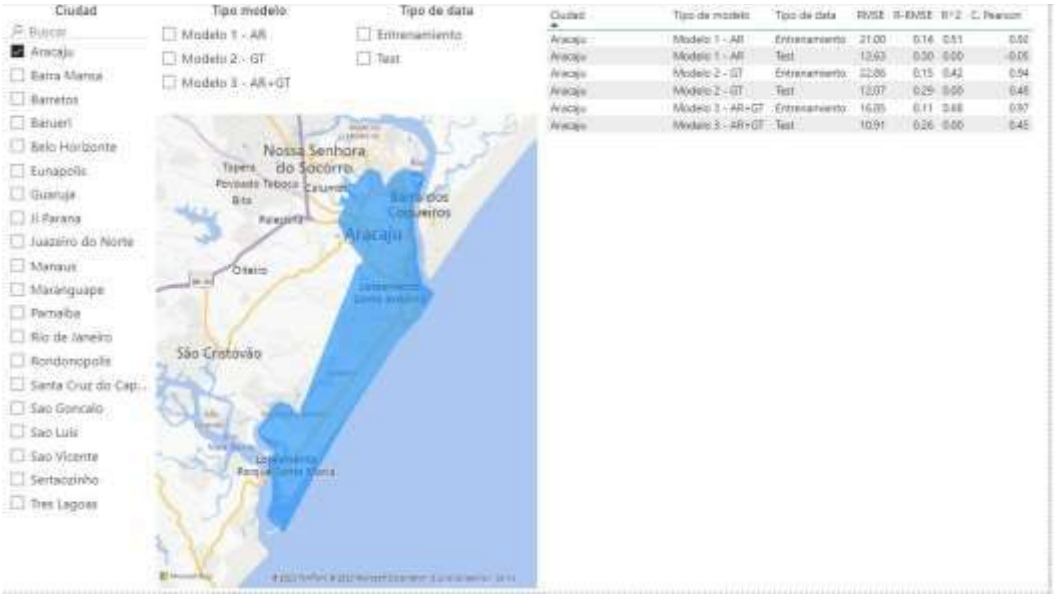


Figura 30

Dashboard general.

La hoja general posee 3 filtros según ciudad, tipo de modelo (AR, GT y AR + GT), y los tipos de datos (entrenamiento o test), junto a un gráfico de la ciudad y una tabla que evidencia los campos mencionados y las métricas de evaluación del modelo.

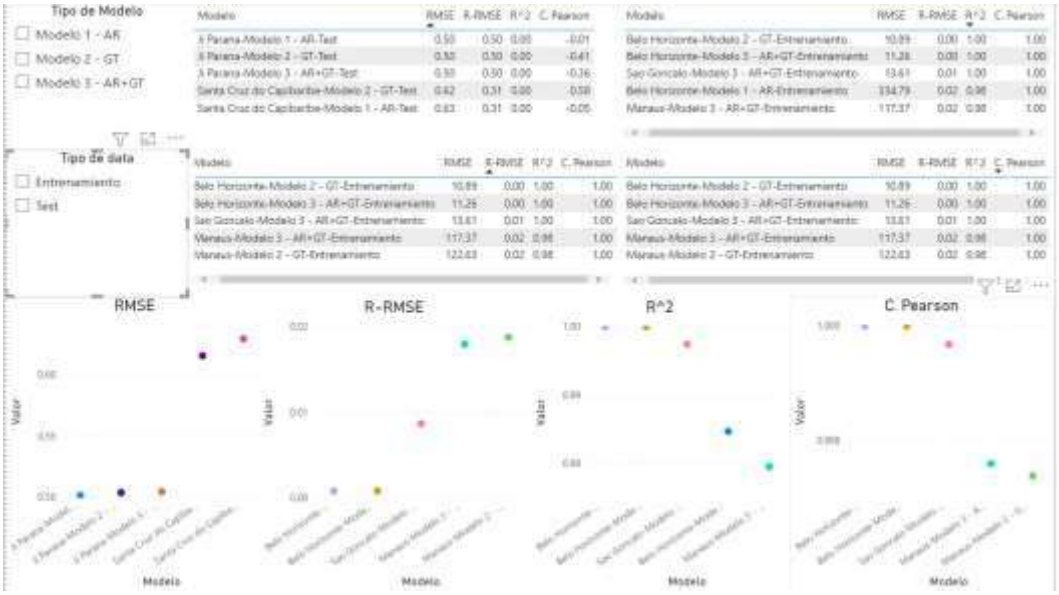


Figura 31

Dashboard mejores 5 métricas por métrica.

La hoja mejores 5 modelos por métrica posee 2 filtros según tipo de modelo (AR, GT y AR + GT), y los tipos de datos (entrenamiento o test), junto a 4 tablas en donde se mencionan los mejores desempeños en relación a determinada métrica con un gráfico de dispersión que lo acompaña.

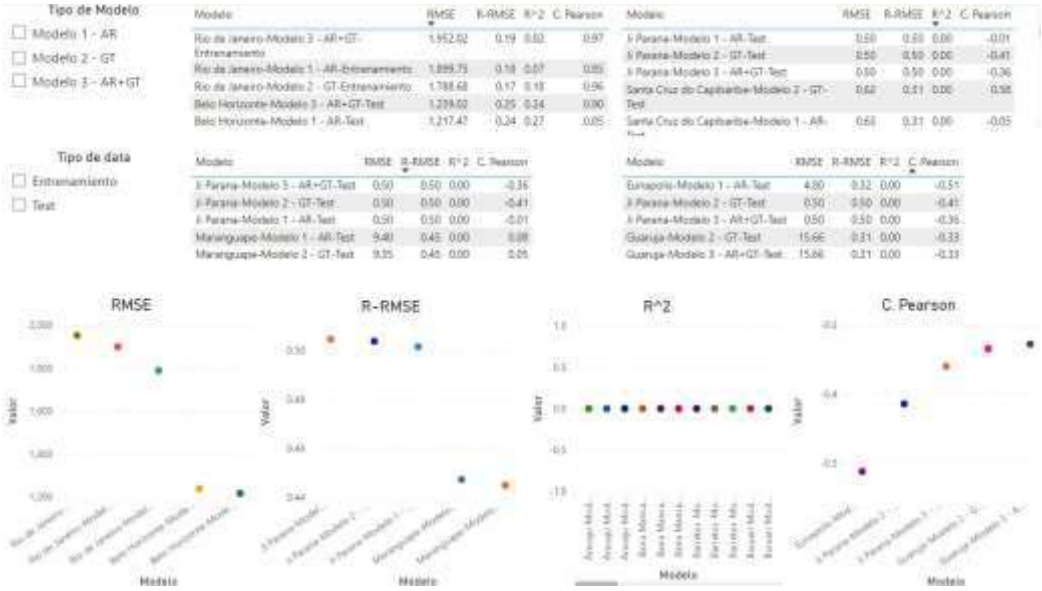


Figura 32

Dashboard peores 5 métricas por métrica.

La hoja peores 5 modelos por métrica es idéntico a la hoja anterior con los peores desempeños en relación a determinada métrica con un gráfico de dispersión que lo acompaña.

En base al dashboard hicimos un resumen de los modelos considerando las 4 métricas de evaluación, para recomendar o no el uso de determinado modelo de predicción de las incidencias del dengue empleando el algoritmo XGBoost:

Tabla 14

Recomendación de modelos en relación a la ciudad

Ciudad	Modelo 1 AR	Modelo 2 GT	Modelo 3 AR+GT	Desempeño (mejor al peor)
Aracaju	Sí	Sí	Sí	Modelo 3, modelo 2, modelo 1
Barra Mansa	Sí	Sí	Sí	Modelo 3, modelo 2, modelo 1
Barretos	No	Sí	No	Modelo 2, modelo 1, modelo 3
Barueri	No	No	No	Modelo 2, modelo 3, modelo 1
Belo Horizonte	No	No	No	Modelo 2, modelo 3, modelo 1
Eunapolis	Sí	Sí	Sí	Modelo 3, modelo 2, modelo 1
Guaruja	Sí	Sí	Sí	Modelo 1, modelo 2, modelo 3
Ji-Parana	No	No	No	Sobreajustamiento

Juazeiro do Norte	Sí	Sí	Sí	Modelo 3, modelo 2, modelo 1
Manaus	No	No	No	Modelo 2, modelo 3, modelo 1
Maranguape	Sí	Sí	Sí	Modelo 3, modelo 2, modelo 1
Parnaiba	Sí	Sí	Sí	Modelo 2, modelo 3, modelo 1
Rio de Janeiro	No	No	No	Modelo 3, modelo 2, modelo 1
Rondonopolis	Sí	Sí	Sí	Modelo 3, modelo 2, modelo 1
Santa Cruz	Sí	Sí	Sí	Modelo 2, modelo 3, modelo 1
Sao Goncalo	No	No	Sí	Modelo 3, modelo 2, modelo 1
Sao Luis	Sí	Sí	Sí	Modelo 1, modelo 2, modelo 3
Sao Vicente	No	Sí	No	Modelo 2, modelo 1, modelo 3
Sertaozinho	No	Sí	No	Modelo 2, modelo 1, modelo 3
Tres Lagoas	Sí	Sí	Sí	Modelo 2, modelo 1, modelo 3

De las 20 ciudades en 15 de ellas se puede al menos emplear un modelo recomendado para predecir la incidencia del dengue empleando el algoritmo XGBoost.

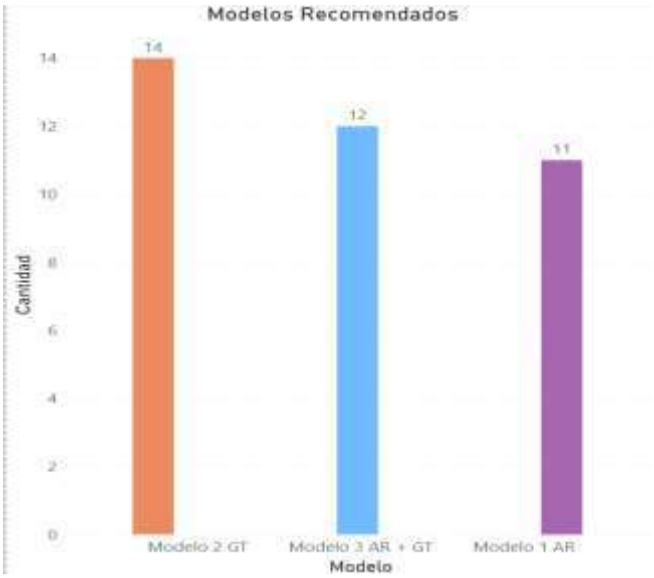


Figura 33

Cantidad de modelos recomendados.

De los modelos recomendados, 14 de ellos son aplicables empleando datos de Google Trends (GT), 12 empleando datos combinados de Google Trends e Históricos, y 11 empleando datos históricos.

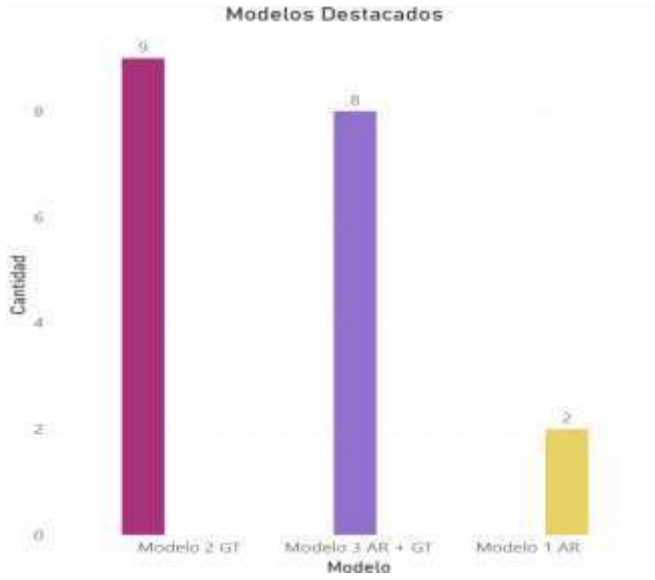


Figura 34

Cantidad de modelos destacados.

De los modelos destacados, 9 lo hicieron empleando data de Google Trends, 8 empleando data combinada de Google Trends e Histórica y 2 empleando datos Históricos.

Los resultados de esta investigación coinciden con (Aiken et al., 2020), que la utilización de búsquedas de Google Trends en relación a una enfermedad confirman un buen desempeño en modelos de machine learning para la predicción de estas y a su vez, demuestra el valor agregado al combinarlos con datos autorregresivos (históricos y en relación al clima).

CONCLUSIONES

1. Conseguimos predecir la incidencia del dengue empleando un modelo de aprendizaje supervisado basado en el algoritmo “Extreme Gradient Boosting” (XGBoost), empleando un conjunto de datos conformado por 272 periodos semanas.

2. Construimos un modelo de aprendizaje supervisado basado en árboles de decisión empleando la data autorregresiva (histórica), con la capacidad de predecir la incidencia del dengue hasta con 14 semanas de anticipación, valiéndonos de la regresión no lineal del algoritmo XGBoost, con buen desempeño para casuísticas donde no existe proporcionalidad entre predictores y el objetivo. Aplicamos técnicas de preprocesamiento para seleccionar características y elección de hiperparámetros, con el fin de mejorar el desempeño del modelo.

3. Empleamos la data de fuentes alternativas confiables como la de Google Trends para afianzar la data histórica y logramos mejorar la precisión de la predicción en nuestro modelo. Adicionalmente, el modelo solamente entrenado con datos de Google Trends, tuvo un mejor desempeño que el modelo entrenado únicamente con datos históricos.

4. Construimos un dashboard para evidenciar los indicadores recopilados por cada modelo para evaluar su desempeño, teniendo como resultado de que nuestros modelos pueden ser aplicados en 15 de las 20 ciudades del dataset, entre ellos destacando los modelos que consumían datos de únicamente Google Trends o una combinación de estos y los datos históricos, demostrando la importancia de la inclusión de datos de fuentes alternativas para afianzar la predicción de las incidencias dengue.

RECOMENDACIONES

1. Para trabajos posteriores, sugerimos obtener y/o recopilar un set de datos con una mayor cantidad de periodos de tiempo y congruentes al lugar en el que se aplica la predicción de las incidencias del dengue, así mismo, un riguroso proceso de detección de anomalías, limpieza y/o normalización de datos para mejorar la calidad de los resultados.

2. Con el fin de un mejor desempeño de los modelos, recomendamos la utilización de diferentes algoritmos de Machine Learning Supervisado que permita captar las series temporales propias de los periodos semanales en el conjunto de datos, empleando indicadores de evaluación afines a esta investigación.

3. Recomendamos explorar diferentes configuraciones de hiperparámetros como la cantidad de árboles a emplearse, profundidad de los árboles y la tasa de aprendizaje, con el propósito de obtener un mejor desempeño del modelo.

4. Para determinar si un modelo tiene un buen rendimiento o no, recomendamos emplear dos o más métricas de evaluación al mismo tiempo e incluso incluir otras métricas no contempladas en esta investigación.

REFERENCIAS BIBLIOGRÁFICAS

Aiken, E. L., McGough, S. F., Majumder, M. S., Wachtel, G., Nguyen, A. T., Viboud, C., & Santillana, M. (2020). Real-time estimation of disease activity in emerging outbreaks using internet search information. *PLOS Computational Biology*, *16*(8), e1008117. https://doi.org/10.1371/journal.pcbi.1008117

Amin, S., Irfan Uddin, M., H. Al-Baity, H., Ali Zeb, M., & Abrar Khan, M. (2021). Machine Learning Approach for COVID-19 Detection on Twitter. *Computers, Materials & Continua*, *68*(2), 2231–2247. https://doi.org/10.32604/cmc.2021.016896

Amin, S., Uddin, M. I., Hassan, S., Khan, A., Nasser, N., Alharbi, A., & Alyami, H. (2020). Recurrent Neural Networks With TF-IDF Embedding Technique for Detection and Classification in Tweets of Dengue Disease. *IEEE Access*, *8*, 131522–131533. https://doi.org/10.1109/ACCESS.2020.3009058

Benedum, C. M., Shea, K. M., Jenkins, H. E., Kim, L. Y., & Markuzon, N. (2020). Weekly dengue forecasts in Iquitos, Peru; San Juan, Puerto Rico; and Singapore. *PLOS Neglected Tropical Diseases*, *14*(10), e0008710. https://doi.org/10.1371/journal.pntd.0008710

Bi, Q., Goodman, K. E., Kaminsky, J., & Lessler, J. (2019). What is Machine Learning? A Primer for the Epidemiologist. *American Journal of Epidemiology*. https://doi.org/10.1093/aje/kwz189

Brett, T. S., & Rohani, P. (2020). Dynamical footprints enable detection of disease emergence. *PLOS Biology*, *18*(5), e3000697. https://doi.org/10.1371/journal.pbio.3000697

Burrell, J. (2016). How the machine 'thinks': Understanding opacity in machine learning algorithms. *Big Data & Society*, *3*(1), 205395171562251. https://doi.org/10.1177/2053951715622512

Cavany, S. M., España, G., Vazquez-Prokopec, G. M., Scott, T. W., & Perkins, T. A. (2021). Pandemic-associated mobility restrictions could cause increases in dengue virus transmission. *PLOS Neglected Tropical Diseases*, *15*(8), e0009603. https://doi.org/10.1371/journal.pntd.0009603

Corvalán, J. G. (2018). Inteligencia artificial: retos, desafíos y oportunidades – Prometea: la primera inteligencia artificial de Latinoamérica al servicio de la

Justicia. *Revista de Investigações Constitucionais*, *5*(1), 295. https://doi.org/10.5380/rinc.v5i1.55334

da Silva Neto, S. R., Tabosa Oliveira, T., Teixeira, I. V., Aguiar de Oliveira, S. B., Souza Sampaio, V., Lynn, T., & Endo, P. T. (2022). Machine learning and deep learning techniques to support clinical diagnosis of arboviral diseases: A systematic review. *PLOS Neglected Tropical Diseases*, *16*(1), e0010061. https://doi.org/10.1371/journal.pntd.0010061

Despotovic, M., Nedic, V., Despotovic, D., & Cvetanovic, S. (2016). Evaluation of empirical models for predicting monthly mean horizontal diffuse solar radiation. *Renewable and Sustainable Energy Reviews*, *56*, 246–260. https://doi.org/10.1016/j.rser.2015.11.058

Elson, W. H., Ortega, E., Kreutzberg-Martinez, M., Jacquerioz, F., Cabrera, L. N., Oberhelman, R. A., & Paz-Soldan, V. A. (2020). Cross-sectional study of dengue-related knowledge, attitudes and practices in Villa El Salvador, Lima, Peru. *BMJ Open*, *10*(10), e037408. https://doi.org/10.1136/bmjopen-2020-037408

Ferdous, M., Debnath, J., & Chakraborty, N. R. (2020). Machine Learning Algorithms in Healthcare: A Literature Survey. *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 1–6. https://doi.org/10.1109/ICCCNT49239.2020.9225642

Ferdousi, T., Cohnstaedt, L. W., & Scoglio, C. M. (2021). A Windowed Correlation-Based Feature Selection Method to Improve Time Series Prediction of Dengue Fever Cases. *IEEE Access*, *9*, 141210–141222. https://doi.org/10.1109/ACCESS.2021.3120309

Feurer, M., & Hutter, F. (2019). *Hyperparameter Optimization* (pp. 3–33). https://doi.org/10.1007/978-3-030-05318-5_1

Fleuren, L. M., Klausch, T. L. T., Zwager, C. L., Schoonmade, L. J., Guo, T., Roggeveen, L. F., Swart, E. L., Girbes, A. R. J., Thoral, P., Ercole, A., Hoogendoorn, M., & Elbers, P. W. G. (2020). Machine learning for the prediction of sepsis: a systematic review and meta-analysis of diagnostic test accuracy. *Intensive Care Medicine*, *46*(3), 383–400. https://doi.org/10.1007/s00134-019-05872-y

Francisco, M. E., Carvajal, T. M., Ryo, M., Nukazawa, K., Amalin, D. M., & Watanabe, K. (2021). Dengue disease dynamics are modulated by the combined influences

of precipitation and landscape: A machine learning approach. *Science of The Total Environment*, *792*, 148406. https://doi.org/10.1016/j.scitotenv.2021.148406

Ho, T.-S., Weng, T.-C., Wang, J.-D., Han, H.-C., Cheng, H.-C., Yang, C.-C., Yu, C.-H., Liu, Y.-J., Hu, C. H., Huang, C.-Y., Chen, M.-H., King, C.-C., Oyang, Y.-J., & Liu, C.-C. (2020). Comparing machine learning with case-control models to identify confirmed dengue cases. *PLOS Neglected Tropical Diseases*, *14*(11), e0008843. https://doi.org/10.1371/journal.pntd.0008843

Hoyos, W., Aguilar, J., & Toro, M. (2021). Dengue models based on machine learning techniques: A systematic literature review. *Artificial Intelligence in Medicine*, *119*, 102157. https://doi.org/10.1016/j.artmed.2021.102157

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, *22*(4), 679–688. https://doi.org/10.1016/j.ijforecast.2006.03.001

Inga-Ávila, M., Churampi-Cangalaya, R., Inga-Aliaga, M., Rodríguez-Giraldez, W., & Vicente-Ramos, W. (2022). Influence of people, processes and technology on business strategy in small enterprise in a Covid 19 environment. *International Journal of Data and Network Science*, *6*(3), 779–786. https://doi.org/10.5267/j.ijdns.2022.3.003

Khan, A. N., Iqbal, N., Rizwan, A., Malik, S., Ahmad, R., & Kim, D. H. (2022). A Criticality-Aware Dynamic Task Scheduling Mechanism for Efficient Resource Load Balancing in Constrained Smart Manufacturing Environment. *IEEE Access*, *10*, 50933–50946. https://doi.org/10.1109/ACCESS.2022.3173157

Koplewitz, G., Lu, F., Clemente, L., Buckee, C., & Santillana, M. (2022). Predicting dengue incidence leveraging internet-based data sources. A case study in 20 cities in Brazil. *PLOS Neglected Tropical Diseases*, *16*(1), e0010071. https://doi.org/10.1371/journal.pntd.0010071

Lalinde, J. D. H., Castro, F. E., Rodr\'\iguez, J. E., Rangel, J. G. C., Sierra, C. A. T., Torrado, M. K. A., Sierra, S. M. C., & Pirela, V. J. B. (2018). Sobre el uso adecuado del coeficiente de correlación de Pearson: definición, propiedades y suposiciones. *Archivos Venezolanos de Farmacolog\'\ia y Terapéutica*, *37*(5), 587–595.

Lam López, J. A., & Alva Arévalo, A. (2021). *Sistema de Registro de Atención a Pacientes (SISRAP) y su relación con los procesos de control en el Hospital II-2*

Tarapoto, 2020 [Info:eu-repo/semantics/bachelorThesis]. Universidad Nacional de San Martín - Tarapoto.

Latif, J., Xiao, C., Imran, A., & Tu, S. (2019). Medical Imaging using Machine Learning and Deep Learning Algorithms: A Review. *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (ICoMET)*, 1–5. https://doi.org/10.1109/ICOMET.2019.8673502

Lee, I., & Shin, Y. J. (2020). Machine learning for enterprises: Applications, algorithm selection, and challenges. *Business Horizons*, *63*(2), 157–170. https://doi.org/10.1016/j.bushor.2019.10.005

Lenhart, A., Castillo, C. E., Villegas, E., Alexander, N., Vanlerberghe, V., van der Stuyft, P., & McCall, P. J. (2022). Evaluation of insecticide treated window curtains and water container covers for dengue vector control in a large-scale cluster-randomized trial in Venezuela. *PLOS Neglected Tropical Diseases*, *16*(3), e0010135. https://doi.org/10.1371/journal.pntd.0010135

Li, C., Wu, X., Wang, X., Yin, J., Zheng, A., & Yang, X. (2021). Ecological environment and socioeconomic factors drive long-term transmission and extreme outbreak of dengue fever in epidemic region of China. *Journal of Cleaner Production*, *279*, 123870. https://doi.org/10.1016/j.jclepro.2020.123870

Liu, Y. E., Saul, S., Rao, A. M., Robinson, M. L., Agudelo Rojas, O. L., Sanz, A. M., Verghese, M., Solis, D., Sibai, M., Huang, C. H., Sahoo, M. K., Gelvez, R. M., Bueno, N., Estupiñan Cardenas, M. I., Villar Centeno, L. A., Rojas Garrido, E. M., Rosso, F., Donato, M., Pinsky, B. A., … Khatri, P. (2022). An 8-gene machine learning model improves clinical prediction of severe dengue progression. *Genome Medicine*, *14*(1), 33. https://doi.org/10.1186/s13073-022-01034-w

McGough, S. F., Clemente, L., Kutz, J. N., & Santillana, M. (2021). A dynamic, ensemble learning approach to forecast dengue fever epidemic years in Brazil using weather and population susceptibility cycles. *Journal of The Royal Society Interface*, *18*(179), 20201006. https://doi.org/10.1098/rsif.2020.1006

Mitchell, R., & Frank, E. (2017). Accelerating the XGBoost algorithm using GPU computing. *PeerJ Computer Science*.

Mudele, O., Frery, A. C., Zanandrez, L. F. R., Eiras, A. E., & Gamba, P. (2021). Modeling dengue vector population with earth observation data and a generalized

linear model. *Acta Tropica*, *215*, 105809. https://doi.org/10.1016/j.actatropica.2020.105809

Muhammad, I., & Yan, Z. (2015). SUPERVISED MACHINE LEARNING APPROACHES: A SURVEY. *ICTACT Journal on Soft Computing*, *05*(03), 946–952. https://doi.org/10.21917/ijsc.2015.0133

Mussumeci, E., & Codeço Coelho, F. (2020). Large-scale multivariate forecasting models for Dengue - LSTM versus random forest regression. *Spatial and Spatio-Temporal Epidemiology*, *35*, 100372. https://doi.org/10.1016/j.sste.2020.100372

Nasteski, V. (2017). An overview of the supervised machine learning methods. *HORIZONS.B*, *4*, 51–62. https://doi.org/10.20544/HORIZONS.B.04.1.17.P05

Natali, E. N., Babrak, L. M., & Miho, E. (2021). Prospective Artificial Intelligence to Dissect the Dengue Immune Response and Discover Therapeutics. *Frontiers in Immunology*, *12*. https://doi.org/10.3389/fimmu.2021.574411

Ochida, N., Mangeas, M., Dupont-Rouzeyrol, M., Dutheil, C., Forfait, C., Peltier, A., Descloux, E., & Menkes, C. (2022). Modeling present and future climate risk of dengue outbreak, a case study in New Caledonia. *Environmental Health*, *21*(1), 20. https://doi.org/10.1186/s12940-022-00829-z

Parra, C., Cernuzzi, L., Rojas, R., Denis, D., Rivas, S., Paciello, J., Coloma, J., & Holston, J. (2020). Synergies Between Technology, Participation, and Citizen Science in a Community-Based Dengue Prevention Program. *American Behavioral Scientist*, *64*(13), 1850–1870. https://doi.org/10.1177/0002764220952113

Pinter, G., Felde, I., Mosavi, A., Ghamisi, P., & Gloaguen, R. (2020). COVID-19 Pandemic Prediction for Hungary; A Hybrid Machine Learning Approach. *Mathematics*, *8*(6), 890. https://doi.org/10.3390/math8060890

Portugal, I., Alencar, P., & Cowan, D. (2018). The use of machine learning algorithms in recommender systems: A systematic review. *Expert Systems with Applications*, *97*, 205–227. https://doi.org/10.1016/j.eswa.2017.12.020

Ratanakomol, T., Roytrakul, S., Wikan, N., & Smith, D. R. (2022). Oroxylin A shows limited antiviral activity towards dengue virus. *BMC Research Notes*, *15*(1), 154. https://doi.org/10.1186/s13104-022-06040-0

Ray, S. (2019). A Quick Review of Machine Learning Algorithms. *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, 35–39. https://doi.org/10.1109/COMITCon.2019.8862451

Robles-Fernández, Á. L., Santiago-Alarcon, D., & Lira-Noriega, A. (2021). American Mammals Susceptibility to Dengue According to Geographical, Environmental, and Phylogenetic Distances. *Frontiers in Veterinary Science*, *8*. https://doi.org/10.3389/fvets.2021.604560

Rodríguez, E. M. (2005). Errores frecuentes en la interpretación del coeficiente de determinación lineal. *Anuario Jur\'\idico y Económico Escurialense*, *38*, 315–331.

Salami, D., Sousa, C. A., Martins, M. do R. O., & Capinha, C. (2020). Predicting dengue importation into Europe, using machine learning and model-agnostic methods. *Scientific Reports*, *10*(1), 9689. https://doi.org/10.1038/s41598-020-66650-1

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., & Haque, U. (2021). Prediction of dengue outbreak in Selangor Malaysia using machine learning techniques. *Scientific Reports*, *11*(1), 939. https://doi.org/10.1038/s41598-020-79193-2

Sanchez-Gendriz, I., de Souza, G. F., de Andrade, I. G. M., Neto, A. D. D., de Medeiros Tavares, A., Barros, D. M. S., de Morais, A. H. F., Galvão-Lima, L. J., & de Medeiros Valentim, R. A. (2022). Data-driven computational intelligence applied to dengue outbreak forecasting: a case study at the scale of the city of Natal, RN-Brazil. *Scientific Reports*, *12*(1), 6550. https://doi.org/10.1038/s41598-022-10512-5

Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, *2*(3), 160. https://doi.org/10.1007/s42979-021-00592-x

Shanker, S. G. (1987). Wittgenstein versus Turing on the nature of Church's thesis. *Notre Dame Journal of Formal Logic*, *28*(4), 615–649.

Sippy, R., Farrell, D. F., Lichtenstein, D. A., Nightingale, R., Harris, M. A., Toth, J., Hantztidiamantis, P., Usher, N., Cueva Aponte, C., Barzallo Aguilar, J., Puthumana, A., Lupone, C. D., Endy, T., Ryan, S. J., & Stewart Ibarra, A. M. (2020). Severity Index for Suspected Arbovirus (SISA): Machine learning for accurate prediction of hospitalization in subjects suspected of arboviral infection.

PLOS Neglected Tropical Diseases, *14*(2), e0007969. https://doi.org/10.1371/journal.pntd.0007969

Song, J., Xie, H., Gao, B., Zhong, Y., Gu, C., & Choi, K.-S. (2021). Maximum likelihood-based extended Kalman filter for COVID-19 prediction. *Chaos, Solitons & Fractals*, *146*, 110922. https://doi.org/10.1016/j.chaos.2021.110922

Souza, C., Maia, P., Stolerman, L. M., Rolla, V., & Velho, L. (2022). Predicting dengue outbreaks in Brazil with manifold learning on climate data. *Expert Systems with Applications*, *192*, 116324. https://doi.org/10.1016/j.eswa.2021.116324

Sun, H., Koo, J., Dickens, B. L., Clapham, H. E., & Cook, A. R. (2022). Short-term and long-term epidemiological impacts of sustained vector control in various dengue endemic settings: A modelling study. *PLOS Computational Biology*, *18*(4), e1009979. https://doi.org/10.1371/journal.pcbi.1009979

Tsantalidou, A., Parselia, E., Arvanitakis, G., Kyratzi, K., Gewehr, S., Vakali, A., & Kontoes, C. (2021). MAMOTH: An Earth Observational Data-Driven Model for Mosquitoes Abundance Prediction. *Remote Sensing*, *13*(13), 2557. https://doi.org/10.3390/rs13132557

Tucto Pinedo, D. K. (2020). *Sistema de vigilancia y control y su influencia en el proceso de la toma de decisiones para el tratamiento del vector Aedes Aegypti en la Dirección Regional de Salud de San Martín* [Info:eu-repo/semantics/bachelorThesis]. Universidad Nacional de San Martín - Tarapoto.

Xu, J., Xu, K., Li, Z., Meng, F., Tu, T., Xu, L., & Liu, Q. (2020). Forecast of Dengue Cases in 20 Chinese Cities Based on the Deep Learning Method. *International Journal of Environmental Research and Public Health*, *17*(2), 453. https://doi.org/10.3390/ijerph17020453

Zhao, N., Charland, K., Carabali, M., Nsoesie, E. O., Maheu-Giroux, M., Rees, E., Yuan, M., Garcia Balaguera, C., Jaramillo Ramirez, G., & Zinszer, K. (2020). Machine learning and dengue forecasting: Comparing random forests and artificial neural networks for predicting dengue burden at national and sub-national scales in Colombia. *PLOS Neglected Tropical Diseases*, *14*(9), e0008056. https://doi.org/10.1371/journal.pntd.0008056

Zheng, X., Zheng, Z., Wu, S., Wei, Y., Luo, L., Zhong, D., & Zhou, G. (2022). Spatial heterogeneity of knockdown resistance mutations in the dengue vector Aedes

albopictus in Guangzhou, China. *Parasites & Vectors*, *15*(1), 156. https://doi.org/10.1186/s13071-022-05241-7

ANEXOS

Anexo 1 Acta de entrevista a la representante de la OGESS Alto Mayo

ACTA DE ENTREVISTA

Siendo las 09.05 am horas del día 10 del mes de Mayo del año 2022, se realizó la entrevista a Rosa del Pilar Saldaña Lino, con el cargo de Jefa de la Unidad Especializada de Epidemiología de la institución OGESS Alto Mayo para conocer el trabajo que realizan en la detección del dengue en San Martín, información que sirve para la elaboración de un proyecto de tesis de la Facultad de Ingeniería de Sistemas e Informática (FISI) de la Universidad Nacional de San Martín (UNSM).

Para dar fe de la entrevista, firmamos:

Entrevistador: Jim Harold Padilla Pierola

DNI: 72167100

Cod. Estudiante: 72167100

Entrevistado: Rosa del Pilar Saldaña Lino

DNI: 72638807

Anexo 2 Recursos en GitHub

https://github.com/JimHPP/indencia_dengue_XGBoost

Modelo de aprendizaje supervisado basado en el algoritmo XGBoost para predicción de la incidencia del dengue

por JIM JAROLD PADILLA PIEROLA

Fecha de entrega: 24-sep-2024 01:41p.m. (UTC-0500)
Identificador de la entrega: 2464325917
Nombre del archivo: XGBoost-JimPadillaPierola-Final.docx (3.84M)
Total de palabras: 15176
Total de caracteres: 85979

Modelo de aprendizaje supervisado basado en el algoritmo XGBoost para predicción de la incidencia del dengue

INFORME DE ORIGINALIDAD

10%	9%	3%	5%
INDICE DE SIMILITUD	FUENTES DE INTERNET	PUBLICACIONES	TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	tesis.unsm.edu.pe Fuente de Internet	3%
2	repositorio.unsm.edu.pe Fuente de Internet	1%
3	Submitted to Universidad Nacional de San Martín Trabajo del estudiante	1%
4	github.com Fuente de Internet	<1%
5	www.csdn.net Fuente de Internet	<1%
6	Submitted to University of East London Trabajo del estudiante	<1%
7	Submitted to Northcentral Trabajo del estudiante	<1%
8	www.coursehero.com Fuente de Internet	<1%