

**Tabla de contenidos**

[Módulo de Aprendizaje de la Máquina (Machine Learning)](about:blank) 0

[Antecedentes 3](#_heading=h.30j0zll)

[Objetivos de la contratación 3](#_heading=h.3znysh7)

[Definición de alcances del producto 4](#_heading=h.2et92p0)

[Objetivos y Alcances 4](#_heading=h.tyjcwt)

[Objetivo general 4](#_heading=h.3dy6vkm)

[Objetivo especifico 5](#_heading=h.1t3h5sf)

[Propuesta conceptual 5](#_heading=h.4d34og8)

[Valor práctico 6](#_heading=h.2s8eyo1)

[Modelos de Aprendizaje Automático o Machine Learning 6](#_heading=h.17dp8vu)

[Definiciones 6](#_heading=h.3rdcrjn)

[Entrenamiento, validación cruzada y verificación 7](#_heading=h.26in1rg)

[Predicción más allá del rango de entrenamiento 8](#_heading=h.lnxbz9)

[Problemas que se resuelven con Aprendizaje Automático 8](#_heading=h.35nkun2)

[Aprendizaje supervisada y no supervisada 8](#_heading=h.1ksv4uv)

[Aplicación de técnicas de Aprendizaje Automático en IPSIM Roya 9](#_heading=h.44sinio)

[Análisis de datos recibidos 10](#_heading=h.2jxsxqh)

[Metodología 12](#_heading=h.z337ya)

[Selección de variables de entrada 13](#_heading=h.3j2qqm3)

[Definición de experimentos 14](#_heading=h.1y810tw)

[Entrenamientos de modelos MLP-ANN 14](#_heading=h.4i7ojhp)

[Selección de variables de entrada 14](#_heading=h.2xcytpi)

[Definición de experimentos 15](#_heading=h.1ci93xb)

[Experimento A 16](#_heading=h.3whwml4)

[Experimento B 16](#_heading=h.2bn6wsx)

[Experimento C 16](#_heading=h.qsh70q)

[Experimento D 16](#_heading=h.3as4poj)

[Entrenamiento de MLP-ANN 16](#_heading=h.1pxezwc)

[Lenguaje interpretativo MATLAB 16](#_heading=h.49x2ik5)

[Lenguaje interpretativo R 17](#_heading=h.2p2csry)

[Resultados 18](#_heading=h.147n2zr)

[Lenguaje interpretativo MATLAB 18](#_heading=h.3o7alnk)

[Lenguaje interpretativo R 19](#_heading=h.23ckvvd)

[Conclusiones 22](#_heading=h.32hioqz)

[Referencias 23](#_heading=h.1hmsyys)

## Antecedentes

El brote de la roya del café que afectó a Centroamérica y República Dominicana desde el ciclo productivo 2012/2013, provocó un serio impacto social en la región, que trajo como consecuencia, la afectación de los ingresos de las familias, debido a que la mayor parte del café que se produce es cultivada por pequeños productores y productoras, con poca capacidad de resiliencia a condiciones adversas directa o indirectamente relacionadas con el cambio y la variabilidad del clima. Ante tal situación, surge el Programa Centroamericano de Gestión Integral de la Roya del Café (PROCAGICA), como una iniciativa internacional que pretende contribuir en la mejora de la situación socioeconómica que enfrenta el sector cafetalero de estos países.

A partir de esta iniciativa financiada por la Unión Europea y coordinada por el Instituto Interamericano de Cooperación para la Agricultura (IICA), se apoyarán los esfuerzos regionales y nacionales para el control integrado de la roya del café, y para reforzar la capacidad de respuesta y adaptación de los pequeños y medianos productores de café ante el cambio y la variabilidad climática.

## Objetivos de la contratación

Esta consultoría constituye la última etapa para la implementación de plataformas automatizadas de recuperación, transmisión, almacenamiento y procesamiento de datos a nivel nacional y regional, con la participación de los productores.

Los países de la consultoría donde el proyecto PROCAGICA tiene influencia, son 5 países principales: Nicaragua, Honduras, El Salvador, Guatemala y República Dominicana, así como 2 países adicionales: Costa Rica y Panamá.

Este trabajo de consultoría tiene como objetivo desarrollar la plataforma informática “Pergamino” con 8 versiones del producto como se desglosa a continuación:

1. Pergamino Regional (1): Sistema de datos regional, que permita el consumo de la información estándar de cada uno de los países[[1]](#footnote-0). Dicho sistema se pretende sea administrado por PROMECAFE.
2. Pergamino país (7): Se desarrollará una versión de la plataforma para cada uno de los países: Guatemala, El Salvador, Honduras, Nicaragua, Costa Rica, Panamá y República Dominicana.

La consultoría deberá suscribir un Contrato de Confidencialidad (NDA) con cada uno de los administradores o gestores del software por país, donde se asegure el manejo confidencial de la información provista por cada uno.

Se desarrollará una implementación de la plataforma que será instalada en cada país que la gestionará. La plataforma básica para cada país será operativa mediante la conexión a las bases de datos públicas. Adicionalmente, dependiendo de las solicitudes expresadas por cada país, la plataforma podrá también funcionar con bases de datos institucionales o privadas.

La plataforma debe tomar en cuenta las herramientas, información y datos existentes, así como mejoras deseables, especialmente considerando la necesidad de integrar variables de diferente naturaleza (biológicas y productivas, físicas, económicas) y de diferentes fuentes para los pronósticos de roya y consecuentes crisis socioeconómicas. La plataforma deberá considerar la posibilidad de calcular en tiempo real diferentes indicadores de riesgo y su combinación para determinar los riesgos de roya y de crisis socioeconómicas, integrando los modelos científicos que se están desarrollando para tal efecto.

También debe contemplar la participación del productor, especialmente a través de los datos de incidencia de roya que él pueda proporcionar, y también a través de recomendaciones personalizadas a partir de los resultados de los modelos. Dicha herramienta o interfaz se embeberá dentro de las herramientas existentes en la medida que las mismas lo permitan. En caso contrario, se acordará con cada uno de los países una interfaz para dicho fin.

## Definición de alcances del producto

El presente informe presenta el desarrollo de un modelo complementario del crecimiento de Roya utilizando técnicas de Machine Learning o aprendizaje automático utilizado para mejorar los resultados de simulación de pronóstico de incidencia obtenidos del modelo IPSIM Roya. Para lo anterior, se propuso un diseño conceptual de la aplicación del corrector de error o modelo complementario utilizando técnicas de Machine Learning, en base a los resultados y desarrollos realizados por el modelo IPSIM Roya, desarrollado por el centro de cooperación internacional en investigación agronómica para el desarrollo (CERAD). Este documento está estructurado de la siguiente forma: Las secciones 2 y 3 presentan en detalle los objetivos del modelo y su propuesta conceptual. Asimismo, se presenta el valor práctico de este desarrollo. En la sección 5 se presenta una breve descripción de los modelos de aprendizaje automático o Machine Learning. En la sección 6 se presenta un análisis preliminar de los datos recibidos de las incidencias observadas y pronosticadas de los países Honduras, El Salvador y República Dominicana. En la sección 7 se presenta la metodología adoptada para entrenar y validar el modelo de corrector de error utilizando técnica de machine learning.

# Objetivos y Alcances

El objetivo general y específico se encuentra a continuación:

## Objetivo general

* Diseñar de forma conceptual un modelo de aprendizaje automático que sea capaz de corregir los errores del pronóstico de incidencia de la roya IPSIM Roya.

## Objetivo específico

* Realizar un análisis a las series temporales de las variables para definir las variables de entrada del modelo de corrección de error.
* Proponer diferentes experimentos de modelos de corrección de error en base a la selección de variables de entrada.
* Implementar el modelo de corrección de errores utilizando técnicas de aprendizaje automático en R Studio y/o MATLAB.
* Elaborar un manual de usuario básico para poder replicar o actualizar los resultados obtenidos.

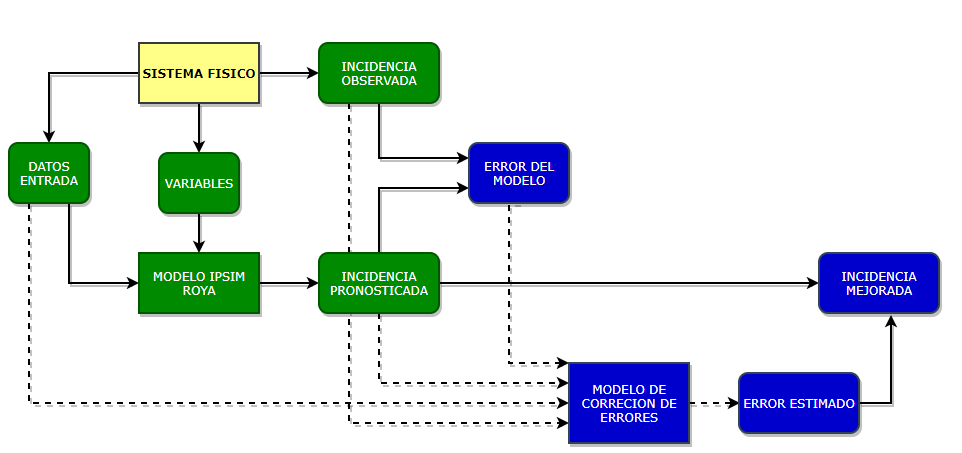
# Propuesta conceptual

En la figura 1 se presenta la propuesta conceptual de la aplicación de corrector de error propuesto. Se puede observar que se parte de los resultados adquiridos del modelo IPSIM Roya, desarrollado por el centro de cooperación internacional en investigación agronómica para el desarrollo (CERAD). En dicho modelo se pronostica las incidencias de crecimiento de la roya en base a la incidencia actual, la lluvia diaria, temperatura media, variedad, carga fructífera, nutrición adecuada, porcentaje de sombra, mes de aplicación de químicos, entre otras. Partiendo de este este modelo se podría mejorar la incidencia de crecimiento de la Roya utilizando cualquiera de estas metodologías

1. Mejorando el modelo IPSIM Roya identificando la fuente de incertidumbre en la predicción de la incidencia. Una vez definido la fuente de incertidumbre se puede evaluar si es posible minimizar esta incertidumbre con mayor adquisición de datos o mejorar la estructura del modelo.
2. Generando un modelo complementario que permita retomar las salidas del modelo IPSIM Roya y los datos de entrada

En esta propuesta se pretende desarrollar el punto número 2 en la generación de un modelo complementario utilizando herramientas de Aprendizaje Automático tal como es Redes Neuronales Artificiales (ANN, por sus siglas en inglés).

Las variables de entrada tales como incidencia, infección, latencia y lavado, se utilizarían como datos de entrada para el modelo complementario basado en técnicas de ANN. Asimismo, el error (incidencia pronosticada – incidencia observada) se utilizaría para pronosticar el mismo error en el siguiente mes, con la finalidad de mejorar la incidencia pronosticada del siguiente mes. Sin embargo, se debe realizar un análisis de selección de variables de entrada para identificar que variables inciden o tienen mayor peso en el pronóstico del error en el modelo de pronóstico de incidencia IPSIM.



*Figura 1. Propuesta conceptual del modelo complementario utilizando técnicas de Machine Learning o Aprendizaje Automático*

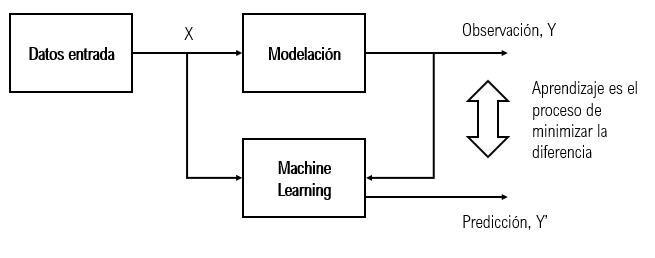
# Valor práctico

El modelo de corrección de error utilizando técnicas de aprendizaje automático (Machine Learning) permitiría mejorar la incidencia pronosticada obtenida del modelo IPSIM Roya. Teóricamente, al minimizarse el error, la incidencia pronosticada estaría más cercana a los valores observados, con lo cual se estaría dando un valor más preciso y exacto de la incidencia futura para la toma de decisiones. Debe tenerse en consideración que los modelos de aprendizaje automático dependen de gran manera de la cantidad de datos con los que se construye, ya que valores fuera del rango con el que fueron “entrenados” o “calibrados”, el modelo tendería a extrapolar resultados, con lo cual se tendría una mayor probabilidad de fallo o desacierto.

# Modelos de Aprendizaje Automático o Machine Learning

## Definiciones

Un método de aprendizaje automático (Machine Learning) es un algoritmo que estima (induce) hasta ahora una dependencia desconocida entre las entrada del sistema y su correspondiente salidas de los datos disponible (Mitchell, 1999). Por dato, se entiende las muestras conocidas que son una combinación de valores de entrada y su correspondiente salida. Si una dependencia es detectada, puede ser utilizada para predecir el futuro del sistema a partir de los valores conocidos.



*Figura 2. Esquema conceptual de un método de aprendizaje automático (Machine Learning)*

La figura 2 muestra el proceso de un aprendizaje automático en base cada instancia (X,Y), el cual trata de “aprender” la función objetivo Y = F(X) que describe como el sistema se comporta. El aprendizaje (o calibración) es el proceso de minimizar la diferencia entre los datos observados y las salidas del modelo. Los datos utilizados para el entrenamiento se denominan serie de entrenamiento. Posterior al proceso de aprendizaje, el modelo de aprendizaje automático es alimentado con nuevas instancias que realizan una predicción.

Uno de los modelos más simples de aprendizaje automático es un modelo de regresión lineal, la cual se construye en base a las observaciones y una fórmula entrenada que permite reproducir aproximadamente a la realidad. Métodos de Machine Learning permiten construir modelos más precisos que modelos lineales.

## Entrenamiento, validación cruzada y verificación

Minimizar la diferencia entre lo observado y lo pronosticado tiene un doble significado. Se debe distinguir

* Minimizar el error durante el proceso de entrenamiento
* Minimizar el error durante el proceso de operación

El modelo que es bien entrenado y con un buen desempeño (error bajo) en la serie de entrenamiento, pero cuando con una nueva instancia no vista por el modelo es ingresada, el error puede ser alto. Esto significa que es necesario tener otros tipos de datos separados, denominado validación cruzada y verificación, el cual no tenga los datos utilizados en la serie de entrenamiento.

Durante el proceso de minimizar el error durante el entrenamiento, se debe tener en cuenta cuando para el proceso para evitar un sobre entrenamiento, es decir, que se genere un sobre ajuste (el modelo trata de seguir todos los puntos de entrenamiento y olvida seguir la tendencia de los datos). En este punto es necesario tener un conjunto de datos para validación cruzada que permita para el proceso de entrenamiento.

Cuando el modelo de aprendizaje automático es puesto en modo operativo, se debe confrontar con una serie de datos que no ha visto previamente, por lo cual se debe verificar su desempeño. En este aspecto se utiliza un tercer conjunto de datos que permita ver el verdadero desempeño del modelo en modo operativo.

## Predicción más allá del rango de entrenamiento

Debido a un modelo de aprendizaje automático aprende lo que se tiene en la serie de entrenamiento, en principio no puede extrapolar. La regla general es tratar de asegurarse que la serie de entrenamiento tenga todos los rangos posibles. Si este es el caso, el modelo podrá aprender el comportamiento del sistema en varias situaciones. Adicionalmente, permitirá extrapolar hasta cierto punto, pero manejando esa predicción con cierto cuidado.

## Problemas que se resuelven con Aprendizaje Automático

De acuerdo con el tipo de salida del sistema, existen dos tipos de problemas que pueden ser resueltos con Aprendizaje Automático: clasificación y regresión. La predicción de valores nominales es denominada clasificación y el algoritmo de Machine Learning es llamado clasificador. Predicción de un valor numérico es llamado regresión.

Un problema de clasificación apunta a identificar las funciones que mapean cualquier entrada dada (X) a una variable nominal (Y). Instancias de entrada para diferentes grupos (también denominados clusters) son asociadas con un valor de salida.

El problema de encontrar los grupos (agrupaciones) en un conjunto de datos de entrada es la agrupación. El agrupamiento es a menudo la fase de preparación para la clasificación: los grupos identificados pueden etiquetarse como clases, cada instancia de entrada puede asociarse con un valor de salida (clase) y el conjunto de instancias {xi, yi} puede construirse. Este conjunto se puede utilizar para entrenar un clasificador. Debido al hecho de que los patrones de entrada no tienen las etiquetas de clase correspondientes y, por lo tanto, dicha clasificación no puede ser dirigida (supervisada) por las etiquetas en los datos de entrenamiento, la agrupación a veces se denomina clasificación no supervisada.

## Aprendizaje supervisada y no supervisada

Anteriormente se definió que un problema de aprendizaje automático puede ser de regresión y de clasificación. Asimismo, el aprendizaje automático puede tener diversos métodos, entre los que destacan los supervisados y no supervisados.

En el aprendizaje supervisado, los algoritmos trabajan con datos “etiquetados”, intentado encontrar una función que, dadas las variables de entrada (input data), les asigne la etiqueta de salida adecuada. El algoritmo se entrena con un “histórico” de datos y así “aprende” a asignar la etiqueta de salida adecuada a un nuevo valor, es decir, predice el valor de salida. (Simeone, 2018). De igual forma, en el aprendizaje supervisado se suele usar problemas de regresión como de clasificación.

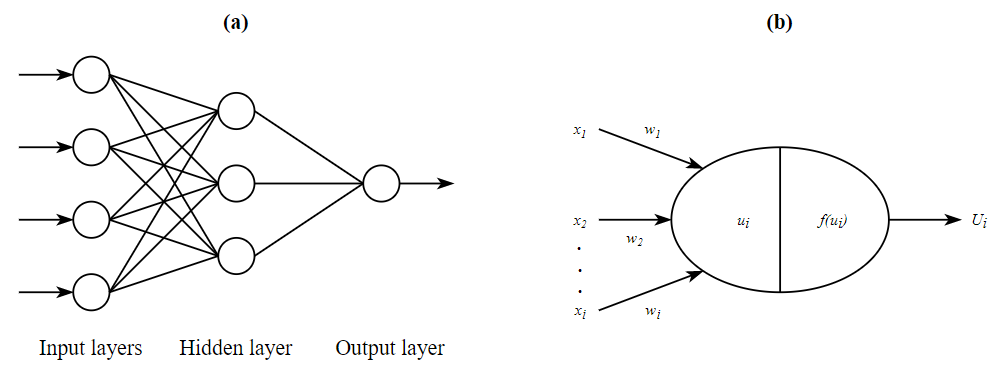
En el aprendizaje no supervisado, el conjunto de entrenamiento consta de entradas sin etiquetar, es decir, de entradas sin ninguna salida deseada asignada. (Simeone, 2018). En el aprendizaje no supervisado se suele usar en problemas de agrupamiento (clustering), perfilado o agrupamientos de co-ocurrencias.

Existe un tercer tipo de aprendizaje el cual es el aprendizaje por refuerzo, que estaría entre un aprendizaje supervisado y no supervisado. Este tipo de aprendizaje se basa en mejorar la respuesta del modelo usando un proceso de retroalimentación.

## Aplicación de técnicas de Aprendizaje Automático en IPSIM Roya

Como se mencionó anteriormente, se pretende utilizar las técnicas de Machine Learning o Aprendizaje Automático como modelo complementario que permita pronosticar el error del modelo IPSIM Roya y mejorar la incidencia pronosticada. Para realizar lo anterior se pretende implementar un aprendizaje supervisado utilizando un algoritmo de Perceptrón Multicapa Redes Neuronales Artificiales (MLP-ANN, por sus siglas en ingles) para un problema de regresión, ya que el valor de salida es un valor numérico.

Una de las principales tecnologías principales para el modelado basado en datos son las redes neuronales artificiales (ANN). Este método se puede utilizar en los campos de agrupación, clasificación y predicción. Las ANN imitan libremente la funcionalidad de las neuronas en el cerebro humano y combinan estas neuronas para reproducir cualquier función multivariable o multivalor (Solomatine, 2016) La figura 3 presenta la arquitectura de la red neuronal artificial (a) y la estructura de una neurona individual (b), que consta de una serie de entradas (x), una función de activación (es decir, una función no lineal) calculada sobre la base de la suma ponderada de todos los entradas y salidas. Esta función de activación puede ser una función logística sigmoidea, gaussiana o tangente hiperbólica. La selección de la función depende del tipo de red y del algoritmo de entrenamiento empleado (Dawson & Wilby, 2001).



*Figura 3. (a) Arquitectura de un perceptrón Multicapa Red Neuronal Artificial y (b) elementos de procesamiento dentro de una neurona o nodo en una ANN.*

La conexión entre neuronas que fluye en una dirección se llama redes de alimentación (feed-foward). Esta dirección de flujo va desde la capa de entrada, a través de capas ocultas que contienen las neuronas, hasta las capas de salida (ver figura 3). Hay dos tipos de redes de retroalimentación que han sido utilizadas por muchos autores: el perceptrón multicapa (MLP) y la función de base radial (RBF). El MLP es definido por Solomatine y Wagener (2011) como un dispositivo que consta de varias capas de neuronas interconectadas entre sí, que transforman las entradas usando una transformación no lineal multiparamétrica para que el modelo resultante sea capaz de aproximar relaciones complejas de entrada y salida. El objetivo del proceso de entrenamiento es optimizar el número de neuronas y capas ocultas a las que se aplica la red mediante un procedimiento de prueba y error (Shamseldin, 1997). Si hay muy pocas neuronas en la capa oculta, es posible que la red neuronal artificial no sea capaz de reproducir la función de salida. Por el contrario, si hay demasiadas neuronas, la ANN puede sobre ajustarse a la función de salida y perder la capacidad de generalizar (Dawson y Wilby, 2001). La función de base radial es una ANN de retroalimentación, similar en topología al MLP (Fernando y Jayawardena, 1998) que intenta aproximar una función y = f (x) mediante una superposición de otras funciones F(x, σ), donde σ es un parámetro que caracteriza el ancho de la función en el espacio de entrada (Solomatine y Wagener, 2011). Normalmente, las funciones gaussianas se utilizan como funciones RBF. Se pueden ver más detalles sobre la red RBF en Dawson y Wilby, 2001; Dayhoff, 1990; Fernando y Jayawardena, 1998; Solomatine y Wagener, 2011; Sudheer y Jain, 2003

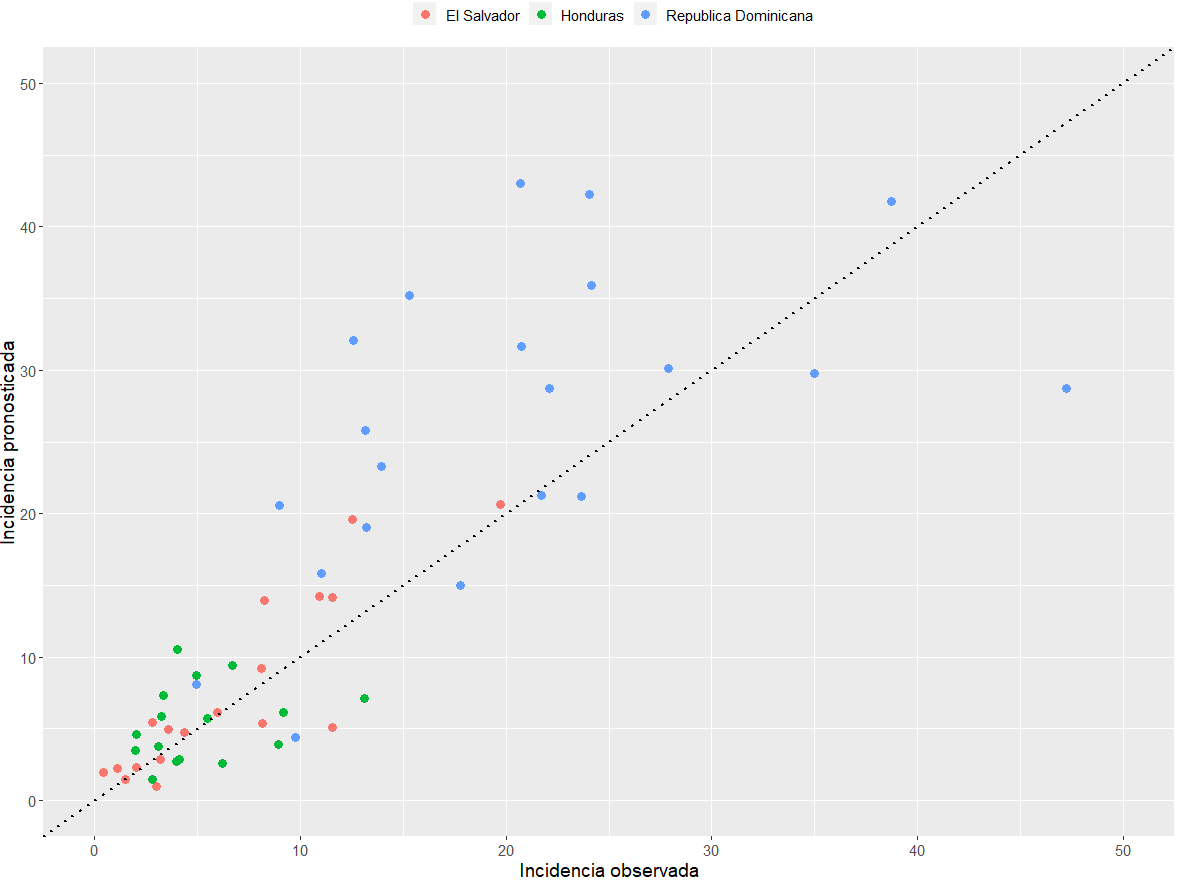
# Análisis de datos recibidos

Se recibieron datos mensuales entre los años 2016 y 2017 correspondiente a los países de El Salvador, Honduras y Republica Dominicana. Entre las variables de entrada y salida del modelo IPSIM Roya recibidas se detalla en la siguiente tabla:

|  |  |
| --- | --- |
| Nombre de variable | |
| Fecha | Nutri\_adecuada |
| País | Sombra |
| Año | Date\_floracion |
| Cat\_altitud | Date\_ini\_cosecha |
| Aparición\_hojas | Date\_fin\_cosecha |
| Infección | Numdia\_floracion |
| Latencia | Numdia\_ini\_cosecha |
| Lavado | Numdi\_fin\_cosecha |
| Efecto\_inc\_defol | Químicos |
| Inoculo | Poda |
| Carga\_fruct | Incidencia observada |
| Variedad | Incidencia pronosticada |

*Tabla 6-1. Variables de entrada y salida recibidas del modelo IPSIM Roya*

Se hizo una verificación preliminar entre la incidencia observada y pronosticada para cada país, los resultados muestran una buena reproducción de la incidencia simulada del modelo IPSIM Roya.

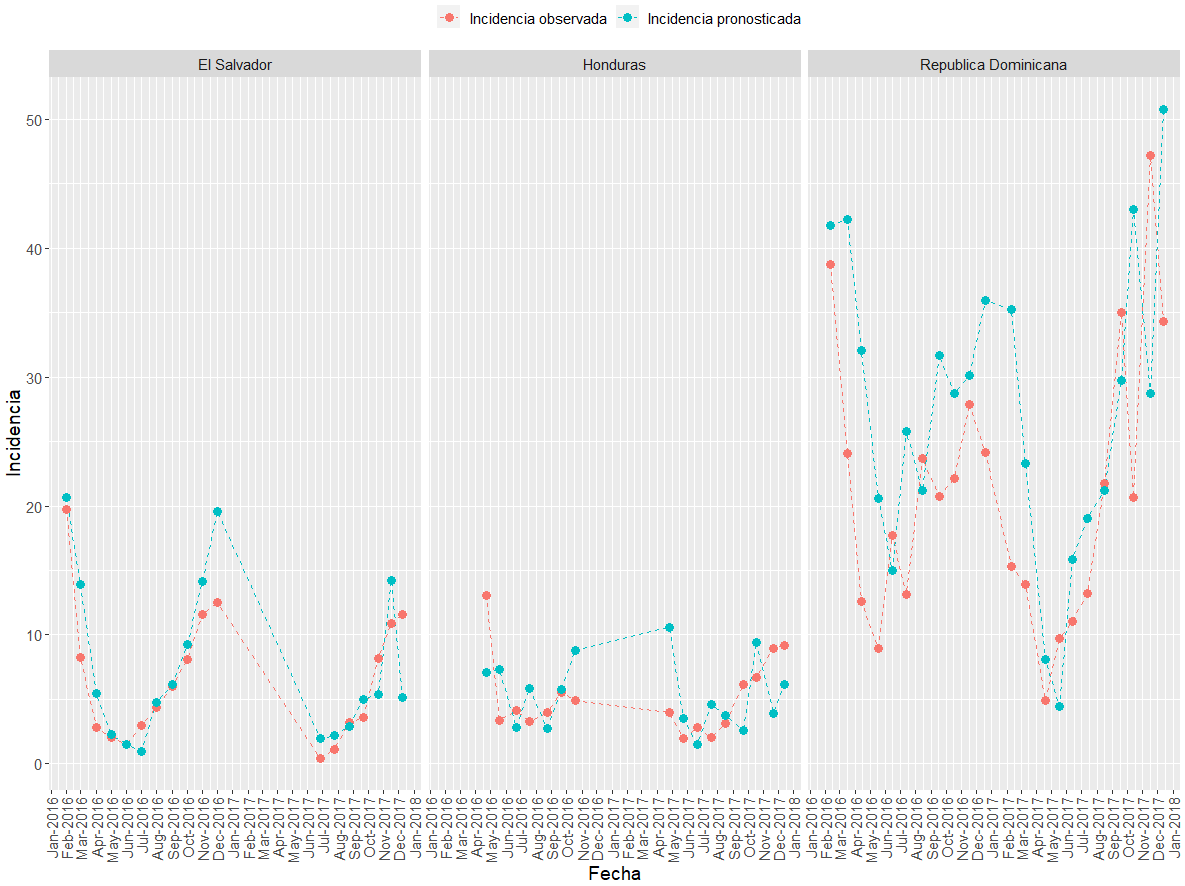


*Figura 4. Verificación visual entre incidencia observada y pronosticada del modelo IPSIM Roya*

De la gráfica anterior se puede comentar lo siguiente:

* Los valores de incidencia pronosticada y simulada para República Dominicana tienen valores diferentes a los que se observan para Honduras y El Salvador
* Las mayores desviaciones entre observaciones y simulaciones se observan para la serie de República Dominicana.
* La mayor cantidad de instancia corresponde a República Dominicana.

Si bien la reproducción de la incidencia por parte del modelo IPSIM Roya es en general muy buena, se observa algunos errores, principalmente en los altos valores de incidencia.



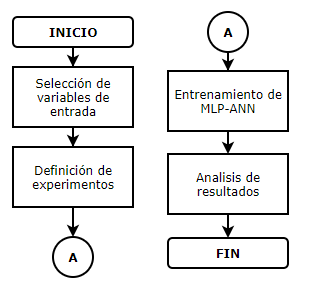
*Figura 5. Incidencia pronosticada y observada por país*

En la figura 5 se observa las series temporales mensuales de la incidencia observada y pronosticada por país. En dicha figura se puede observar meses con periodos faltantes, principalmente en la serie de El Salvador y Honduras, siendo la serie de República Dominicana la más completa. Debido a lo anterior, se decidió trabajar con la serie correspondiente a República Dominicana para la construcción del modelo corrector de error utilizando herramientas de aprendizaje automático. Sin embargo, la serie de República Dominicana no tiene un número de instancia (14) suficiente para construir un modelo basado en datos robustos, ya que los modelos de aprendizaje automático son modelos que dependen de la cantidad de instancias.

En síntesis, se recibieron en total 56 instancias correspondiente a la serie temporal de incidencia de los países de El Salvador, Honduras y República Dominicana. La serie de Republica Dominicana es la serie más completa que se tiene registro, ya que solamente posee un mes faltante entre los años 2016 y 2017.

# Metodología

Para la aplicación de la técnica de Aprendizaje Automático utilizando Redes Neuronales Artificiales (MLP-ANN) como corrector de error en las simulaciones del modelo IPSIM Roya, se realizaron las siguientes actividades mostrada en la siguiente figura



*Figura 6. Metodología desarrollada para implementar modelo complementario utilizando técnicas de Machine Learning*

## Selección de variables de entrada

Una de las primeras tareas a realizar es definir las variables a utilizar como dato de entrada para construir la Red Neuronal Artificial (MLP-ANN). La elección de la variable del modelo que debe considerar la física del proceso subyacente. En Machine Learning, la física se introduce mediante la elección de la variable de entrada relevante que contribuye a predecir el valor de salida. Bowden et al. (2005) indicó que muchos modeladores a menudo presentan una gran cantidad de entradas y dependen de la ANN para identificar las entradas. Esta deficiencia en el proceso de modelado puede aumentar la complejidad, reducir el proceso de aprendizaje y el desempeño del modelo.

En este estudio, las variables apropiadas para estimar el error del modelo IPSIM Roya fue utilizando la Información Mutua Promedio (AMI, en inglés). La información Mutua Promedio (Shannon, 1948) se utiliza para investigar la relación lineal, no lineal y los efectos de retardo entre las variables de series de tiempo. Adicionalmente, es la medida de información disponible de un conjunto de datos que tiene conocimiento de otro conjunto de datos. En otras palabras, representa la medida de información que se puede aprender de un conjunto de datos (X) teniendo el conocimiento de otro conjunto de datos (Y).

|  |  |
| --- | --- |
|  | (1) |

La ecuación 1 presenta la fórmula de información mutua promedio donde PX(x) y Py(y) son las funciones de densidad de probabilidad marginal de X e Y, respectivamente; y PXy(x,y) es la función de densidad de probabilidad conjunta de X e Y. Un valor alto de AMI indicaría una fuerte dependencia entre dos variables. Por el contrario, si X es completamente independiente de Y, entonces AMI es igual a cero.

En este punto se analizarán todas las variables numérica de entrada que utiliza el modelo IPSIM Roya, con la finalidad de identificar las que tenga un valor alto de AMI. Asimismo, se evaluará el retardo de las variables ya que puede existir una relación entre el error y dicha variable que permita mejorar la selección de variables de entrada.

## Definición de experimentos

En base al análisis de variables de entrada detallado anteriormente, se generarán cuatro (4) experimentos con diferentes combinaciones de variables de entrada para pronosticar el error. La formulación general quedaría de la siguiente forma

|  |  |
| --- | --- |
|  | (2) |

En donde L es el retardo óptimo para la variable X, Y o Z. Debe tenerse en consideración que las variables X, Y y Z pueden ser alguna de las variables detalladas en la tabla 1.

## Entrenamientos de modelos MLP-ANN

En el proceso de entrenamiento de la MLP-ANN se realizará las siguientes actividades

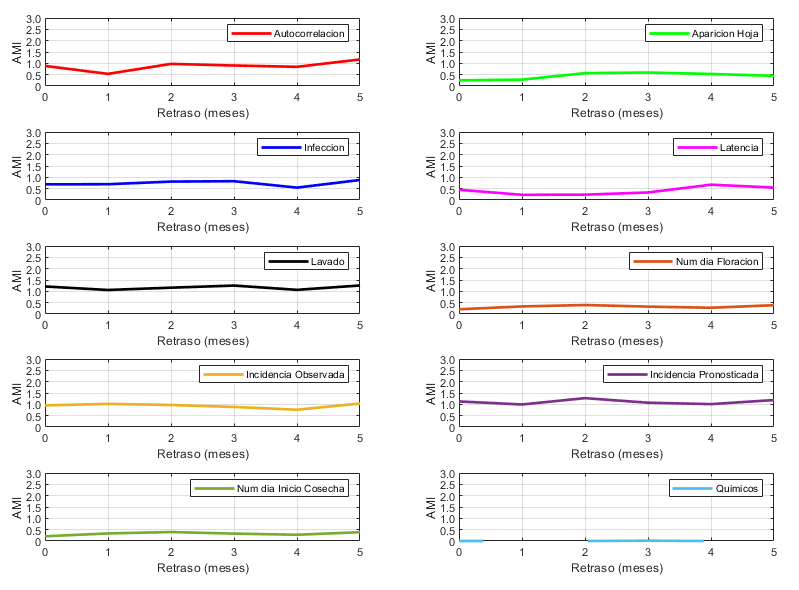
* Se separará los datos de República Dominicana en dos conjuntos de datos: entrenamiento (mayor número de instancias) y de validación cruzada. Adicionalmente se dividió una pequeña porción de datos para verificación del modelo.
* Se generarán scripts computacionales en lenguajes interpretativos MATLAB y R. La diferencia de aplicación de técnicas de Machine Learning entre ambos lenguajes radica en el hecho de que MATLAB ofrece una mejor optimización del aprendizaje de la Red Neuronal Artificial.
* En el proceso de construcción de la MLP-ANN se debe calibrar el número óptimo de neuronas o nodos. Este proceso se realiza de forma manual, tomando como consideración que el número óptimo de neuronas es cuando la raíz del error cuadrático medio (RMSE, por sus siglas en ingles. Ver ecuación 3) entre la incidencia observada (Yobs) y la incidencia simulada (Ysim) aumenta en la serie de validación cruzada.
* Una vez definido el número de neuronas optimas por cada experimento, se verifica el modelo con el set de datos de verificación.
* El experimento que brinde un error menor en la serie de verificación se propondrá como piloto para el modelo de Machine Learning como corrector de error.

|  |  |
| --- | --- |
|  | (3) |

# Selección de variables de entrada

El proceso de selección de variables de entrada permite identificar las variables que brindan más información en la estimación del error pronosticado. Asimismo, se utilizó la autocorrelación del error pasado con el pronosticado para evaluar si existe una relación (lineal o no línea) en la misma variable.

Como se mencionó en la etapa de metodología se utilizó la Información Mutua Promedio (AMI) como indicador de la relación lineal o no lineal entre las variables. En la figura 7 se observa el AMI entre el error pronosticado y las diferentes variables de entrada del IPSIM Roya y el error mismo. Se puede observar valores de AMI cercano a 1, lo que indica que existe una relación entre las variables en consideración. Entre dichas variables que muestra un alto valor de AMI se selecciona las variables de **infección, lavado, la incidencia observada, la incidencia pronosticada y el error mismo (autocorrelación).**



*Figura 7.Información Mutua Promedio (AMI) entre el error pronosticado y las diferentes variables del modelo IPSIM Roya.*

Si bien todas las variables seleccionadas corresponden a un retraso igual a 0 (mes actual), no indica que no se puedan utilizar el resto de los retrasos para identificar otras variables. Para ilustrar lo anterior, se observa en la figura 7 que la relación de la Latencia y el Error Pronosticado para un retraso igual de 4, el AMI es cercano a 1. Se desconoce la razón por la cual la latencia de cuatro meses atrás incide en el error pronosticado en el siguiente mes del modelo. Es por esta razón que no se utilizó esta variable en la definición de experimentos.

# Definición de experimentos

Las variables seleccionadas en la sección 8 de este documento fueron retomadas para definir diferentes experimentos para entrenar la Red Neuronal Artificial (MLP-ANN). Los cuatros experimentos son mostradas a continuación:

## Experimento A

|  |  |
| --- | --- |
|  | (4) |

## Experimento B

|  |  |
| --- | --- |
|  | (5) |

## Experimento C

|  |  |
| --- | --- |
|  | (6) |

## Experimento D

|  |  |
| --- | --- |
|  | (7) |

Como se puede observar en las ecuaciones 4, 5, 6 y 7 correspondiente a cada experimento, las variables seleccionadas en la actividad de selección de variables de entrada corresponden a las variables actuales o de retardo igual a 0. Los experimentos A, B y C utilizan las variables de entrada del modelo IPSIM Roya, además de el error actual del modelo. El experimento D solamente utiliza las variables de entrada del modelo IPSIM Roya.

# Entrenamiento de MLP-ANN

El proceso de entrenamiento de una Red Neuronal Artificial (MLP-ANN) se realizó en base a las variables de entrada de los cuatros experimentos definidos en los apartados anteriores y con la utilización de herramientas computacionales como MATLAB y R. En los siguientes apartados se describe el proceso de entrenamiento para cada lenguaje interpretativo utilizado.

## Lenguaje interpretativo MATLAB

MATLAB ofrece el paquete Machine Learning el cual tiene diferentes herramienta para desarrollar aprendizaje supervisado y no supervisado. Para el proceso de entrenamiento utilizando Redes Neuronales Artificiales de cada experimento, se generaron las siguientes funciones y scripts específico para este proyecto.

* Función *“MLPfunction.m”* que permite entrenar una Red Neuronal Artificial y obtener un objeto con la Red Neuronal Artificial entrenada.
* Script *“mlp-incidencia”* que permite ejecutar la función *“MLPfunction.m”* y realizar múltiples iteraciones con entrenamientos.

El proceso de entrenamiento se modificaron los números de neuronas de forma manual con el fin de identificar el número óptimo de nodos o neuronas para cada experimento. Dentro de cada entrenamiento con un número de neuronas específico, se realizaron más de 1000 corridas con la finalidad de remover condiciones iniciales aleatorias que tienen este tipo de modelos. En síntesis, el proceso de entrenamiento para cada experimento que se realizó con MATLAB fue el siguiente:

* Se inicia con un número de neuronas igual a 2
* Se ejecuta el script “mlp-incidencia.m” y automáticamente se ejecuta un proceso de entrenamiento para ese número de neuronas definido anteriormente. En el proceso de entrenamiento se han definidos los siguientes criterios correspondiente a la arquitectura de la red neuronal artificial
  + Función de transferencia “log sigmoid” en la capa escondida.
  + Función lineal en la capa de salida
  + Función de optimización “Levenberg-Marquardt”
* Se estima el error en la serie de validación cruzada
* Se verifica si el error en la validación cruzada se ha disminuido. En este sentido pueden suceder dos condiciones
  + Si en el caso que el error en la validación cruzada haya aumentado, entonces se detiene el proceso de entrenamiento y el número óptimo de neuronal es el que se está evaluando actualmente.
  + Si el error en la validación cruzada ha disminuido, entonces se aumenta un número de neuronas.

## Lenguaje interpretativo R

Adicionalmente a MATLAB, el lenguaje interpretativo R ofrece muchos paquetes para técnicas de Machine Learning. La ventaja de este lenguaje de programación es que es de uso gratuito. Sin embargo, la desventaja específica con los paquetes de Machine Learning es que existen muchos paquetes con diferentes objetivos específicos, los cuales pueden generar confusión. Para el proceso de Redes Neuronales Artificiales se realizó un script computacional en R utilizando el paquete “neuralnet”.

En el proceso de entrenamiento para cada uno de los experimentos se realizan las siguientes tareas:

* importación de cada variable de entrada y salida de cada experimento.
* normalización de las variables de entrada y salida en base al máximo y mínimo de cada variable.
* Separación de los datos en entrenamiento y validación cruzada
* Se definen los siguientes criterios de arquitectura de la Red Neuronal Artificial:
  + Función de activación en la capa escondida “Log Sigmoid”
  + Función lineal en la capa de salida
  + Algoritmo de entrenamiento: “Backpropagation”
* Se inicia con un número de neuronas igual a 2
* Se ejecuta el script correspondiente para el experimento deseado.
* Se verifica si el error en la validación cruzada se ha disminuido. En este sentido pueden suceder dos condiciones
  + Si en el caso que el error en la validación cruzada haya aumentado, entonces se detiene el proceso de entrenamiento y el número óptimo de neuronal es el que se está evaluando actualmente.
  + Si el error en la validación cruzada ha disminuido, entonces se aumenta un número de neurona.

# Resultados

En los siguientes apartados se presentan los resultados de la Raíz del Error cuadrático Medio (RMSE, por sus siglas en inglés) de las series de entrenamiento y validación cruzada, la cual fue elaborada para cada experimento desarrollado en MATLAB y R.

## Lenguaje interpretativo MATLAB

#### Experimento A

Basado en los resultados obtenidos en la tabla 11-1, se puede observar que el error en la serie de validación aumenta cuando se define tres neuronas. Por lo tanto, la arquitectura de la Red Neuronal Artificial para este experimento tiene un número de neurona óptimo de dos

|  |  |  |
| --- | --- | --- |
| Neuronas | Entrenamiento | Validación cruzada |
| 2 | 1.51 | 2.53 |
| 3 | 2.09 | 4.04 |

*Tabla 11-1. Raíz del error cuadrático medio (RMSE) de la serie de entrenamiento y validación cruzada en el Experimento-A*

#### Experimento B

Basado en los resultados obtenidos en la tabla 11-2, se puede observar que el error en la serie de validación aumenta cuando se define siete neuronas. Por lo tanto, la arquitectura de la Red Neuronal Artificial para este experimento tiene un número de neurona óptimo de seis.

|  |  |  |
| --- | --- | --- |
| Neuronas | Entrenamiento | Validación cruzada |
| 2 | 1.92 | 4.83 |
| 3 | 1.77 | 4.50 |
| 4 | 0.00 | 4.03 |
| 5 | 0.00 | 3.65 |
| 6 | 0.00 | 2.82 |
| 7 | 0.00 | 4.03 |

*Tabla 11-2. Raíz del error cuadrático medio (RMSE) de la serie de entrenamiento y validación cruzada en el Experimento-B*

#### Experimento C

Basado en los resultados obtenidos en la tabla 11-3, se puede observar que el error en la serie de validación aumenta cuando se define seis neuronas. Por lo tanto, la arquitectura de la Red Neuronal Artificial para este experimento tiene un número de neurona óptimo de cinco

|  |  |  |
| --- | --- | --- |
| Neuronas | Entrenamiento | Validación cruzada |
| 2 | 5.06 | 5.24 |
| 3 | 1.77 | 4.48 |
| 4 | 0.00 | 3.52 |
| 5 | 0.00 | 2.95 |
| 6 | 0.00 | 2.98 |

*Tabla 11-3. Raíz del error cuadrático medio (RMSE) de la serie de entrenamiento y validación cruzada en el Experimento-C*

#### Experimento D

Basado en los resultados obtenidos en la tabla 11-4, se puede observar que el error en la serie de validación aumenta cuando se define seis neuronas. Por lo tanto, la arquitectura de la Red Neuronal Artificial para este experimento tiene un número de neurona óptimo de cinco.

|  |  |  |
| --- | --- | --- |
| Neuronas | Entrenamiento | Validación cruzada |
| 2 | 4.98 | 5.06 |
| 3 | 3.47 | 4.82 |
| 4 | 0.00 | 4.35 |
| 5 | 0.00 | 3.53 |
| 6 | 0.00 | 4.55 |

*Tabla 11-4. raíz del error cuadrático medio (RMSE) de la serie de entrenamiento y validación cruzada en el Experimento-D*

Entre los cuatro experimentos, se puede observar que el experimento que reduce el error en la serie de validación cruzada es el experimento A (Ver tabla 11-5).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Experimento A** | **Experimento B** | **Experimento C** | **Experimento D** |
| RMSE | 2.53 | 2.82 | 2.95 | 3.53 |

*Tabla 11-5. Raíz del error cuadrático medio (RMSE) para cada experimento en la serie de validación cruzada.*

## Lenguaje interpretativo R

#### Experimento A

Basado en los resultados obtenidos en la tabla 11-6, se puede observar que el error en la serie de validación aumenta cuando se define cuatro neuronas. Por lo tanto, la arquitectura de la Red Neuronal Artificial para este experimento tiene un número de neurona óptimo de tres.

|  |  |  |
| --- | --- | --- |
| Neuronas | Entrenamiento | Validación cruzada |
| 2 | 1.16 | 9.30 |
| 3 | 0.47 | 9.10 |
| 4 | 0.43 | 9.50 |

*Tabla 11-6. Raíz del error cuadrático medio (RMSE) de la serie de entrenamiento y validación cruzada en el Experimento-A*

#### Experimento B

Basado en los resultados obtenidos en la tabla 11-7, se puede observar que el error en la serie de validación aumenta cuando se define cinco neuronas. Por lo tanto, la arquitectura de la Red Neuronal Artificial para este experimento tiene un número de neurona óptimo de cuatro.

|  |  |  |
| --- | --- | --- |
| Neuronas | Entrenamiento | Validación cruzada |
| 2 | 1.60 | 15.27 |
| 3 | 0.68 | 8.18 |
| 4 | 0.91 | 7.64 |
| 5 | 0.27 | 8.59 |

*Tabla 11-7. Raíz del error cuadrático medio (RMSE) de la serie de entrenamiento y validación cruzada en el Experimento-B*

#### Experimento C

Basado en los resultados obtenidos en la tabla 11-8, se puede observar que el error en la serie de validación aumenta cuando se define seis neuronas. Por lo tanto, la arquitectura de la Red Neuronal Artificial para este experimento tiene un número de neurona óptimo de cinco.

|  |  |  |
| --- | --- | --- |
| Neuronas | Entrenamiento | Validación cruzada |
| 2 | 2.19 | 9.55 |
| 3 | 0.77 | 8.56 |
| 4 | 0.62 | 8.30 |
| 5 | 0.25 | 7.02 |
| 6 | 0.84 | 8.53 |

*Tabla 11-8. Raíz del error cuadrático medio (RMSE) de la serie de entrenamiento y validación cruzada en el Experimento-C*

#### Experimento D

Basado en los resultados obtenidos en la tabla 11-9, se puede observar que el error en la serie de validación aumenta cuando se define cinco neuronas. Por lo tanto, la arquitectura de la Red Neuronal Artificial para este experimento tiene un número de neurona óptimo de cuatro.

|  |  |  |
| --- | --- | --- |
| Neuronas | Entrenamiento | Validación cruzada |
| 2 | 2.31 | 8.38 |
| 3 | 1.65 | 7.22 |
| 4 | 1.34 | 6.89 |
| 5 | 0.96 | 7.91 |

*Tabla 11-9. Raíz del error cuadrático medio (RMSE) de la serie de entrenamiento y validación cruzada en el Experimento-D*

Entre los cuatro experimentos, se puede observar que el experimento que reduce el error en la serie de validación cruzada es el experimento D (Ver tabla 11-10).

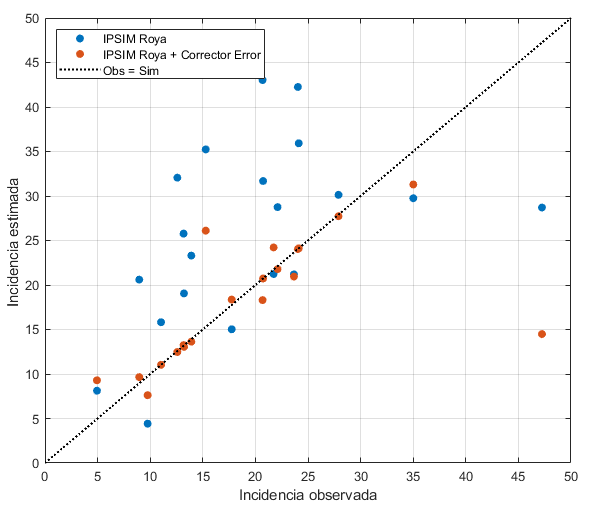
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Experimento A** | **Experimento B** | **Experimento C** | **Experimento D** |
| RMSE | 9.10 | 7.64 | 7.02 | 6.89 |

*Tabla 11-10. Raíz del error cuadrático medio (RMSE) para cada experimento en la serie de validación cruzada.*

Comparando los resultados obtenidos entre MATLAB y R (tabla 11-5 y 11-10), se puede observar que el método utilizado en MATLAB reduce significativamente el error en la serie de validación cruzada. Por lo tanto, se sugiere utilizar el experimento A obtenido a través de MATLAB para utilizarlo de forma operativa.

Una vez definido el experimento adecuado como modelo complementario para pronosticar el error del IPSIM Roya, se realizó una verificación para identificar cuánto mejora la salida con este corrector de error. Cabe mencionar nuevamente que el modelo de Machine Learning realiza una predicción del error en el próximo mes, por lo que para calcular la incidencia mejorada se debe realizar el siguiente calculo.

|  |  |
| --- | --- |
|  | (4) |



*Figura 8. comparación entre los resultados del IPSIM Roya (azul) y el IPSIM Roya + Corrector de error (Rojo)*

Los resultados obtenidos con la serie de datos utilizada para construir el modelo se muestran en la figura 8, en donde se puede observar que el modelo complementario de corrector de error reduce el error del modelo IPSIM Roya. Sin embargo, existen algunas instancias que el modelo tiende a empeorar el resultado. Esto podría deberse a la limitante de datos que existe actualmente y que indicaría que el modelo es solamente válido en los rangos con los que se entrenó en las variables de entrada y salida.

# Conclusiones

Entre las conclusiones encontradas se detallan la siguientes:

* Para la generación de un modelo corrector de errores utilizando técnicas de Machine Learning, se observó que podría existir un relación entre las variables infección, lavado, la incidencia observada, la incidencia pronosticada en el error del modelo IPSIM Roya. Sin embargo, la cantidad de datos no fueron los suficientes como para llegar a un resultado concluyente.
* En el proceso de entrenamiento se utilizó diferentes experimentos, los cuales fueron implementados en MATLAB y R. Comparando los resultados obtenidos entre MATLAB y R (tabla 11-5 y 11-10), se puede observar que el método utilizado en MATLAB reduce significativamente el error en la serie de validación cruzada. De igual forma, se observa que el experimento A brinda mejores resultados que los experimentos restantes. Por lo tanto, se sugiere utilizar el experimento A, obtenido a través de MATLAB para utilizarlo de forma operativa.
* Se puede observar que el modelo complementario de corrector de error reduce el error del modelo IPSIM Roya. Sin embargo, existen algunas instancias que el modelo tiende a empeorar el resultado. Esto podría deberse a la limitante de datos que existe actualmente y que indicaría que el modelo es solamente válido en los rangos con los que se entrenó en las variables de entrada y salida.
* Los resultados del modelo no son concluyentes ya que existe una limitante del número de datos o instancias utilizados para entrenar y verificar el modelo de corrección de error MLP-ANN. Se recomienda aumentar el número de instancias y reentrenar los modelos de corrección de error cuando se tenga suficientes datos de incidencia observada y pronosticada en República Dominicana.

# Referencias

Bowden, G. J., Dandy, G. C., & Maier, H. R. (2005). Input determination for neural network models in water resources applications. Part 1—background and methodology. *Journal of Hydrology*, *301*(1–4), 75–92. https://doi.org/10.1016/j.jhydrol.2004.06.021

Dawson, C. W., & Wilby, R. L. (2001). Hydrological modelling using artificial neural networks. *Progress in Physical Geography: Earth and Environment*, *25*(1), 80–108. https://doi.org/10.1177/030913330102500104

Dayhoff, J. E. (1990). *Neural network architectures: an introduction*. Van Nostrand Reinhold Co.

Fernando, D. A. K., & Jayawardena, A. W. (1998). Runoff forecasting using RBF networks with OLS algorithm. *Journal of Hydrologic Engineering*, *3*(3), 203–209.

Mitchell, T. M. (1999). Machine learning and data mining. *Communications of the ACM*, *42*(11), 30–36.

Shamseldin, A. Y. (1997). Application of a neural network technique to rainfall-runoff modelling. *Journal of Hydrology*, *199*(3–4), 272–294.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, *27*(3), 379–423.

Simeone, O. (2018). A Very Brief Introduction to Machine Learning With Applications to Communication Systems. *ArXiv:1808.02342 [Cs, Math]*. Retrieved from http://arxiv.org/abs/1808.02342

Solomatine. (2016). Data-driven modelling: machine learning, data mining and knowledge discovery.

Solomatine, D., & Wagener, T. (2011). 2.16 Hydrological Modeling, 24.

Sudheer, K. P., & Jain, S. K. (2003). Radial basis function neural network for modeling rating curves. *Journal of Hydrologic Engineering*, *8*(3), 161–164.

1. Se entiende como información estándar, la información acordada y estandarizada por los países para el análisis situacional de los riesgos asociados al café [↑](#footnote-ref-0)