

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

Facultad de Ingeniería

PREDICCIÓN DE DATOS FALTANTES EN CONJUNTOS DE DATOS CLIMATOLÓGICOS MEDIANTE K-NN

Fecha: Marzo 2026

Tesis presentada para obtener el grado de:

Licenciatura en Ingeniería Geofísica

Presenta

Estefanía Espinosa Fernández

Directores de tesis

Víctor Manuel Vázquez Báez

Gabriela Yáñez Pérez

Contenido

Agradecimientos **3**
Resumen **4**
Introducción **5**
Capítulo 1 ..... **7**
1.1 Planteamiento del problema 7
1.2 Justificación 7
1.3 Objetivos 8
1.3.1 Objetivo general 8
1.3.2 Objetivos específicos 8
1.4 Hipótesis 9
1.4.1 Alcance y limitaciones 9
1.4.1.1 Alcance 9
1.4.1.2 Limitaciones 9
Capítulo 2. Marco Teórico **11**
2. 1 Datos faltantes 11
2.1.1 Tratamiento de datos faltantes 11
2.1.1.1 Métodos de imputación 12
2.2 Inteligencia Artificial y Machine Learning 13
2.2.3 Conjuntos de entrenamiento y prueba 14
2.2.4 Principales problemas en Machine Learning 15
2.2.5 Preprocesamiento de datos 16
2.2.6 Tipos de aprendizaje 17
2.3 k-NN 18
2.3.1 Hiperparámetros de k-NN 19
2.4 Evaluación del modelo 20
2.4.1 Error cuadrático medio 20
2.4.2 Raíz cuadrada del error cuadrático medio 21
2.4.4 Normalización 21
2.5 Inteligencia artificial en Python 22
Capítulo 3. Metodología **23**
3.1 Adquisición y preprocesamiento de datos. 23
3.2 Procesamiento y selección del modelo 23
3.3 Aplicación del modelo 24
Resultados **24**
Conclusiones **35**
Referencias **36**

Agradecimientos

A mi familia: mis padres, mi hermana, mis perritas y mis abuelos, por ser mi mayor motivación y apoyo. A mis directores de tesis por acompañarme y guiarme en cada parte del proceso.

El personal autor agradece al Laboratorio Nacional de Supercómputo del Sureste de México (LNS), que pertenece al padrón de laboratorios nacionales SECIHTI, por los recursos computacionales, el apoyo y la asistencia técnica brindados, a través del proyecto No. 202302038C y 202601007C.

The autors thankfully acknowledge computer resources, technical advice and support provided by Laboratorio Nacional de Supercómputo del Sureste de México (LNS), a member of the SECIHTI national laboratories, with project No. 202302038C y 202601007C.

Resumen

La presente investigación explora la posibilidad de emplear inteligencia artificial para rellenar datos nulos en conjuntos climatológicos. El propósito es mejorar la correlación entre precisión y necesidades de cómputo con respecto a los métodos convencionales. Algunos análisis, como modelado de acuíferos, predicción del clima, caracterización de cuerpos de agua, entre otros, requieren bases de datos de datos climático, y aunque estas son de acceso libre en México, cuentan con espacios vacíos que necesitan ser rellenados. Aunque existen algunos métodos de imputación, estas soluciones tradicionales suelen comprometer la precisión o el tiempo del estudio. Por lo tanto, se propone como alternativa al algoritmo k-NN, que consulta puntos cercanos al de interés para definir un valor, y que destaca por su adaptabilidad y rapidez de procesamiento.

Se comparó el error obtenido mediante k-NN con los errores que resultan al utilizar uno de los métodos de tratado de nulos más comunes (imputación por medidas de tendencia central). El método k-NN demostró ser una mejor aproximación.

Introducción

Los datos climatológicos son de gran importancia para múltiples análisis, que van desde planeación urbana, hasta análisis de cambio climático. Esta información se obtiene mediante mediciones que se almacenan en bases de datos. No obstante, debido a fallas en los aparatos de medición, o del propio almacenamiento, es poco común encontrar conjuntos íntegros; en prácticamente todos los casos, suelen encontrarse datos nulos o con valores inválidos.

Múltiples estudios requieren de conjuntos completos, por lo que el tratamiento de nulos es un paso fundamental en el preprocesamiento. Por lo tanto, es común que múltiples analistas busquen la manera óptima de rellenar estos espacios vacíos. Algunos optan por métodos convencionales, como rellenar con medidas de tendencia central. Otros, como Duarte et.al. (2022), han explorado la posibilidad de rellenar utilizando información calculada a través de microondas y observaciones infrarrojas satelitales, comparando este método con modelos de regresión que, los autores indican, son ampliamente utilizadas en la resolución de este problema. Otros autores proponen el uso de algoritmos de Inteligencia Artificial, como Park et.al. (2023), que utilizaron un perceptrón (un modelo neuronal simple) para brindar una solución alternativa. Distintos métodos conllevan ventajas y desventajas, comprometiendo en ocasiones el tiempo o los recursos de cómputo por mayor precisión y viceversa.

La Inteligencia Artificial (IA) comienza a perfilarse como una alternativa cada vez más explorada para resolver este tipo de conflictos. No solo por la gran cantidad de algoritmos que engloba, permitiendo analizar el problema desde distintos enfoques, sino porque ofrece la posibilidad de automatizar los procesos, permitiendo generar un sistema fácilmente adaptable a estos conjuntos. Uno de los primeros

algoritmos que se han utilizado para predecir valores mediante IA es k-NN, que no por eso ha dejado de ser vigente. Este algoritmo consiste en predecir valores utilizando k puntos más cercanos al faltante. Su gran flexibilidad, precisión, simplicidad y capacidad de análisis, lo convierten en una herramienta con un enorme potencial para el rellenado de datos nulos, especialmente para aquellos conjuntos donde los datos siguen alguna tendencia.

Entonces, esta investigación explora la posibilidad de emplear el algoritmo k-NN para rellenar datos nulos de las bases de datos de normales climatológicas de la Comisión Nacional del Agua para la Ciudad de Puebla, buscando un método que equilibre precisión con simplicidad, con un uso bajo de recursos y tiempo de cómputo. El análisis toma en cuenta cuatro atributos: precipitación (mm), temperatura máxima (C), temperatura mínima (C) y evaporación (mm). Cada una fue analizada individualmente, ya que el estudio también busca proveer un método capaz de funcionar para cada atributo sin información adicional.

El primer capítulo presenta el planteamiento del problema, la justificación, los objetivos, la hipótesis y los alcances del proyecto. El segundo capítulo, explora conceptos teóricos valiosos para este trabajo, principalmente relacionados a la Inteligencia Artificial y al algoritmo k-NN. En el tercer capítulo se describe la metodología seguida para realizar la investigación. Y, por último, los resultados y conclusiones presentan los hallazgos, junto a una breve comparación con un método común de rellenado de nulos: imputación por medidas de tendencia central.

Capítulo 1

1.1 Planteamiento del problema

La mayoría de los métodos y algoritmos para analizar fenómenos (incluyendo el modelado de acuíferos o implementación de algoritmos de Inteligencia Artificial), requieren conjuntos de datos completos, pero en la práctica es común encontrar valores nulos. Se han propuesto un gran número de soluciones a este problema, entre las que destacan el rellenado con medidas de tendencia central (debido a su bajo coste computacional), Radio Normal (que utiliza la información de otras estaciones, e interpolaciones. No obstante, estos métodos, o ignoran patrones sutiles e involucran pérdida de información, comprometiendo la integridad de ciertos estudios, o requieren de información adicional y altos costos computacionales, convirtiéndose en soluciones ineficientes que requieren de una significativa intervención y adaptación al aplicarse a nuevos conjuntos. Es entonces que proveer un método que tome en cuenta las tendencias de los datos, que pueda adaptarse fácilmente a nuevos conjuntos y cuyo costo computacional sea razonable es fundamental para realizar análisis de manera eficiente, simple y precisa.

1.2 Justificación

Poseer un conjunto de datos de calidad es una parte fundamental para cualquier análisis y los estudios climatológicos no son una excepción. No obstante, lo más común es encontrar datos imperfectos que requieren tratamiento. Como una solución, este estudio busca presentar a la Inteligencia Artificial como una herramienta auxiliar que, sin requerir de grandes inversiones de tiempo y recursos, es capaz de proporcionar un rellenado de datos nulos que detecta tendencias locales, incrementando la precisión de los datos de entrada y, como consecuencia, de los análisis realizados a partir de ellos. El trabajo se enfoca en el tratamiento

aplicado a conjuntos obtenidos de la base de datos de la CONAGUA y, aunque presenta un enfoque hacia la Ciudad de Puebla, busca ser de utilidad para métodos realizados a partir de conjuntos similares, convirtiéndose en una referencia en la utilización de estos métodos en territorio Mexicano.

Con este propósito, se propone k-NN como un método de imputación para conjuntos de datos climatológicos, el algoritmo presenta ventajas como su capacidad de adaptarse a distintos tipos de conjuntos y su rapidez de cálculo. De esta manera, el producto será un código que requiera poca participación de un usuario humano para aplicarlo a conjuntos nuevos y sea adaptable a otros tipos de conjuntos similares, mejorando la precisión con respecto a otros métodos sin sacrificar las necesidades de cómputo.

1.3 Objetivos

1.3.1 Objetivo general

Generar un código computacional que impute datos climatológicos empleando el algoritmo de inteligencia artificial k-NN incluido en la librería Scikit-Learning de Python, para que este pueda convertirse en una alternativa a métodos de imputación tradicionales que sea precisa sin aumentar considerablemente el tiempo de cómputo.

1.3.2 Objetivos específicos

1. Investigar las características de un sistema de inteligencia artificial con el algoritmo k-NN, prestando atención a sus necesidades de formato para implementarlo en Python de forma eficiente.
2. Generar un código computacional capaz de preprocesar los datos climatológicos, transformándolos a un formato legible por los módulos de

inteligencia artificial y buscando la optimización para que sea fácilmente aplicable a otros conjuntos.

3. Realizar cálculos de error, comparando los valores generados por k-NN con los obtenidos por imputación con medidas de tendencia central, con la finalidad de asegurar que sea una mejor aproximación que los métodos tradicionales.

1.4 Hipótesis

El algoritmo k-NN representará una alternativa viable a los métodos de rellenado de datos convencionales para conjuntos climatológicos, convirtiéndolo en una mejor aproximación sin aumentar considerablemente el tiempo de cómputo.

1.4.1 Alcance y limitaciones

1.4.1.1 Alcance

- El código es capaz de comparar de forma cuantitativa la precisión y requisitos computacionales de la imputación mediante k-NN con los métodos tradicionales para conjuntos de datos climatológicos.
- Debido al diseño del código, es posible sustituir los conjuntos de datos seleccionados por cualquier otro conjunto del mismo tipo.
- El código puede implementarse a conjuntos de datos de otro tipo modificando pocas líneas de código.

1.4.1.2 Limitaciones

- Se requieren datos históricos para poder imputar un dato.

- El algoritmo es ineficiente con conjuntos de datos demasiado grandes debido a necesidades de memoria del algoritmo (con menos de diez mil datos no ocurre).
- No obstante que se trata de un código de aplicación universal el modelo debe ser entrenado para cada nuevo conjunto de datos cada vez.

Capítulo 2. Marco Teórico

2. 1 Datos faltantes

Los datos faltantes o nulos son espacios vacíos o con valores no válidos dentro de un conjunto. No es extraño encontrar conjuntos con datos nulos, esto puede ocasionarse por múltiples factores: medidas no registradas, errores en la medición, valores imposibles, entre otros. Muchos algoritmos y operaciones computacionales requieren un conjunto con valores numéricos bien definidos en cada instancia para poder llevar a cabo el procesamiento. Así, el tratamiento de valores faltantes no solo es una de las primeras etapas en análisis de conjuntos, sino un procedimiento fundamental para garantizar la calidad de los resultados. Existen distintos métodos para llevar a cabo este proceso, y su selección debe ser cuidadosa, considerando las características particulares del conjunto y de cada valor ausente, así como el tipo de análisis.

2.1.1 Tratamiento de datos faltantes

Uno de los métodos de tratamiento de nulos más sencillos es litwise, o análisis con datos completos, que consiste en eliminar las observaciones con valores faltantes y trabajar solo con las que están completas. Si la ausencia está asociada a datos que poseen alguna característica en común se puede generar un sesgo. Adicionalmente, al eliminar demasiadas observaciones ocurren grandes pérdidas de información. (Zas, 2023)

Zas (2023), también describe al análisis con datos disponibles, que consiste en dividir los datos en subconjuntos. Por ejemplo, si se tienen algunas observaciones con seis variables y otras con cinco, se realiza un análisis para las de seis y otro para las de cinco. Una desventaja es que no se trabaja con los mismos datos para

generar modelos, y los costos (computacionales, de capital humano y financiero, o tiempo) se elevan al trabajar con más de un conjunto.

Por otro lado, están los métodos de imputación, en los que se rellena el faltante con algún valor. Cada método para generar valores sintéticos posee sus propias ventajas y desventajas.

2.1.1.1 Métodos de imputación

Imputación por mediana: Si los valores no son completamente aleatorios puede causar sesgo ante la ausencia predominante de datos de cierto tipo (como litwise). Además, solo es funcional con variables cuantitativas.

Imputación por media: Tiene las mismas desventajas que la mediana. Además, es sensible a valores extremos.

Imputación por moda: funciona como los dos otros métodos (y posee tendencia al sesgo), sustituyendo faltantes con la moda, la ventaja es que es funcional con variables cualitativas y cuantitativas.

Imputación mediante modelos de regresión: se genera un modelo de regresión para rellenar faltantes con los valores predichos por este. Este modelo puede ser problemático ya que genera sesgos al afectar las correlaciones entre variables o atributos, esto puede solucionarse agregando una medida de ruido aleatoria a cada valor rellenado.

Imputación cold deck: Se rellenan con valores externos, de conjuntos similares, o incluso mediciones previas del mismo objeto de estudio.

Imputación hot deck: Utiliza el valor de observaciones similares para rellenar los datos faltantes, las observaciones pueden elegirse de forma aleatoria, pueden agruparse observaciones según sus características y seleccionar observaciones de

ese grupo, escoger las observaciones más cercanas mediante una medida de distancia, e incluso en variables temporales se puede imputar el valor anterior o posterior.

Imputación múltiple: Se obtienen múltiples valores para un solo faltante, se consiguen n conjuntos con los valores completos, y luego se combinan esos conjuntos. (Zas, 2023)

2.2 Inteligencia Artificial y Machine Learning

Tradicionalmente se considera que la inteligencia artificial se refiere a todas las técnicas que buscan que las máquinas piensen y aprendan como seres humanos, pero se ha buscado redefinirla de forma objetiva, por lo que se puede decir que la inteligencia artificial engloba al conjunto de técnicas que buscan que las máquinas se comporten o piensen de forma “racional”. (AWS, 2017)

Sus aplicaciones son tan amplias que van desde propósitos de entretenimiento hasta apoyo en áreas como la medicina (con imágenes médicas), pasando por las finanzas (con predicción de valores), modelos de lenguaje, generación y clasificación de imágenes, entre otros, y constantemente surgen nuevos usos para estas herramientas, aplicables en prácticamente cualquier área del conocimiento.

Ya que la Inteligencia Artificial engloba a tantas técnicas y utilidades, se pueden distinguir subconjuntos dentro de esta área, y uno de los más importantes es el Machine Learning, que engloba al conjunto de algoritmos y modelos que los sistemas utilizan para realizar tareas complejas para las que no han sido programados explícitamente. (AWS, 2017)

2.2.3 Conjuntos de entrenamiento y prueba

Existen muchos algoritmos de Machine Learning y cada uno aprende de forma diferente, algunos se inspiran en los seres vivos, como random trees, que se inspira en los árboles, generando soluciones aleatorias representadas por “hojas”, que “votan” por el valor más probable, o las redes neuronales, que se inspiran en las neuronas de los seres humanos. A pesar de esta diversidad, los algoritmos de Machine Learning tienen algo en común: aprenden de un conjunto de datos y generan un modelo para predecir valores para conjuntos distintos.

Para entrenar un modelo de Machine Learning, los datos se dividen en dos: conjunto de entrenamiento y conjunto de prueba. Se aprende de datos que funcionan como “ejemplo”, que es el conjunto de entrenamiento, y una vez que se entrena al sistema y se genera un modelo, se evalúa su desempeño en el conjunto de prueba, que simula cómo se comportaría dicho modelo en nuevas instancias.

Los datos suelen dividirse en 80% para el conjunto de entrenamiento y 20% para el conjunto de prueba, aunque la proporción depende del número y tipo de datos y el problema que se quiera resolver. Es posible seleccionar los datos para cada conjunto de forma arbitraria, pero este procesamiento es tedioso y poco recomendable, ya que esta selección puede generar sesgo. La mayoría de las paqueterías para trabajar con Machine Learning incluyen módulos que dividen aleatoria y automáticamente el conjunto. Además, existen métodos que permiten seleccionar instancias representativas de ciertos atributos, en caso de ser necesario.

2.2.4 Principales problemas en Machine Learning

Aun cuando los sistemas de Machine Learning son capaces de realizar tareas de forma prácticamente autónoma, el preprocesamiento es responsabilidad de quien realiza el análisis, y es en este momento donde suelen surgir los primeros errores. Géron (2019), destaca la importancia de tener un buen conocimiento del conjunto de datos y del tipo de modelo a generar, ya que una incorrecta selección de algoritmo, o malas elecciones en el número y tipo de datos, pueden conducir a alguno de los problemas más comunes en los modelos generados con estas herramientas: sobreajuste y subajuste.

- El sobreajuste significa que el modelo funciona muy bien para los datos con los que se entrenó, pero no con datos nuevos, ya que se realizan falsas generalizaciones o patrones que se presentan de forma particular para este conjunto, esto generalmente se relaciona con un número insuficiente de datos.
- El subajuste, por otro lado, implica que el modelo no es una buena solución para el problema, y ni siquiera en los datos de entrenamiento presenta un buen ajuste, esto puede ser ocasionado por una mala elección del modelo, porque los atributos no son suficientes, o los que hay no son significativos, entre otras causas.

Estos problemas pueden prevenirse, pero suelen identificarse hasta las fases finales del modelado, ya que se vuelven evidentes al evaluar con el conjunto de prueba, y corregir sobre esta evaluación conduce a sobreajuste. Una solución propuesta en el libro de Géron (2019), es la validación cruzada, que permite elaborar y evaluar distintos modelos sin recurrir al conjunto de prueba.

2.2.5 Preprocesamiento de datos

Los conjuntos deben tener una estructura específica para ser analizados: cada dato (al que también se llama instancia) está asociado a una lista de características llamadas atributos. Un ejemplo de esto es la relación entre el precio de un inmueble y su ubicación, el número de pisos, el número de habitaciones, los metros cuadrados que ocupa, etc. El precio es la instancia y el resto de las características son sus atributos; el número de atributos es arbitrario, yendo desde el uno hasta un número ilimitado, aunque es importante considerar el costo computacional y los errores a los que puede conducir un número demasiado grande o una incorrecta selección de atributos.

Entonces, el conjunto de datos engloba a un determinado número de instancias con sus características, tomando en cuenta que los atributos seleccionados deben estar presentes en todos los casos (Retomando el ejemplo anterior, es imposible asociar la influencia de la ubicación en el precio si solo un dato la tiene como atributo y los demás no la indican). (Géron, 2019)

La manera más simple de comprender el arreglo de los datos es imaginando una tabla, con los atributos como columnas y las instancias como filas (aunque hay algoritmos con estructuras de entrada distintas, este es el caso más simple y, por lo tanto, más fácil de comprender). Cuando se acomodan los datos de esta manera es más sencillo llevar a cabo el preprocesamiento.

En resumen, el preprocesamiento consiste en acomodar los datos al formato de entrada requerido por cada algoritmo, ajustando el número de instancias y de atributos para optimizar el sistema, asegurándose de que cada dato sea legible y válido, incluyendo un tratamiento de datos nulos, que dependerá de las características de cada problema.

2.2.6 Tipos de aprendizaje

Cuando los datos se encuentran correctamente ordenados es más sencillo analizarlos, y entonces seleccionar qué tipo de algoritmo es conveniente aplicar. Dependiendo del problema, el sistema puede aprender de distintas maneras. Según la participación de un analista externo, Gerón (2019) describe cuatro clases:

- Aprendizaje supervisado: Se tienen “etiquetas” que son, básicamente, la respuesta esperada al problema. Por ejemplo, si se tiene una serie de imágenes de perros, gatos y conejos, las etiquetas serían, para cada imagen “perro”, “gato” y “conejo”, cada etiqueta está asociada a atributos, que en este caso son la imagen.
- Aprendizaje no supervisado: No se agregan etiquetas, solo los atributos y se espera que el sistema aprenda de ellos, por ejemplo, si se tiene una serie de imágenes de perros, gatos y conejos, la IA busca similitudes y diferencias entre las imágenes para asignarles una clase, distinguir por sí misma cuántas se parecen lo suficiente para pertenecer a una misma especie.
- Aprendizaje semi-supervisado: Algunos datos poseen etiquetas y otros no. Por ejemplo, de las fotos de perros, gatos y conejos, algunos están etiquetados, lo que puede ser un punto de partida para que aprenda el sistema, pero posteriormente comienza a aprender a realizar una clasificación sin que los datos de entrenamiento tengan etiquetas.
- Aprendizaje por refuerzo: Se premia o castiga al sistema cuando acierta o se equivoca al resolver un problema. Por ejemplo, si clasifica a la imagen de un perro como “gato”, se castiga; si clasifica a un conejo como conejo, se premia.

También se pueden clasificar los algoritmos según las características de los datos de entrada, en Imam et. al. (2024) se describen dos tipos:

- Aprendizaje paramétrico: Los datos de entrada tienen una función de distribución de probabilidad específica, realizan inferencias sobre la forma del modelo de ajuste. Así, el número de parámetros está definido desde el inicio y el número de datos no influye en este. Estos algoritmos suelen ser más rápidos. Un ejemplo de este tipo de algoritmos es regresión lineal.
- Aprendizaje no paramétrico: Los datos no cumplen con una función de distribución de probabilidad específica. Al contrario que el aprendizaje paramétrico, no hace inferencias sobre la forma del modelo, lo que proporciona flexibilidad. El costo y tiempo computacionales suelen ser proporcionales al número de datos. Un ejemplo de este tipo de algoritmos es k-NN.

2.3 k-NN

Son las siglas de k-Nearest Neighbors, es decir, k-vecinos más cercanos. Como su nombre lo indica, es un algoritmo de inteligencia artificial que consulta a determinado número (k) de instancias cercanas al punto que se investiga para asignarle una clase o valor.

Es un algoritmo de aprendizaje supervisado no paramétrico y también se considera un algoritmo “perezoso”, esto último significa que se realiza todo el cálculo al hacer una predicción, es decir, no genera un modelo interno, sino que lleva a cabo la consulta cada vez que se investiga un nuevo dato. (Alemán-Rivas

et.al., 2015) Debido a lo anterior, el algoritmo requiere una cantidad importante de memoria para funcionar. (IBM, 2022)

Según Du et.al. (2022), el algoritmo se puede resumir en tres pasos primordiales:

1. Definir las distancias entre el punto de investigación y los de consulta.
2. Encontrar los k vecinos cercanos.
3. Realizar la clasificación o predicción.

2.3.1 Hiperparámetros de k-NN

En Machine Learning, se denomina hiperparámetro a las posibles modificaciones en un algoritmo para que genere un modelo. Es importante distinguirlos de los parámetros porque algunos algoritmos (como regresión lineal) generan funciones matemáticas con sus propios parámetros; estos son propios del modelo (no todos los algoritmos generan modelos con parámetros). Por el contrario, los hiperparámetros son propios de cada algoritmo y es a partir de ellos que se generará el modelo. Por ejemplo, en los sistemas con redes neuronales, se ajusta el número de neuronas.

Para k-NN es necesario definir tres hiperparámetros: k, el tipo de distancia y el peso. Como el algoritmo requiere calcular distancias para identificar los vecinos cercanos, este es un factor importante para definir. El tipo de distancia más común es la euclidiana (la línea recta entre dos puntos), pero es posible utilizar otros tipos, como la Manhattan (que se mueve a través de un grillado). (Davi, 2023)

El peso se refiere a qué tanta influencia tiene cada vecino sobre el punto de consulta. Hay dos tipos de peso: el uniforme, en el que todos los vecinos tienen la misma influencia, y el ponderado, en el que la influencia depende de la distancia que cada punto tiene con el de consulta.

K es el número de vecinos, y no existen reglas bien definidas para establecerlo, aunque sí hay algunas recomendaciones: k debe ser impar para evitar empates entre clases; en relación con el número de datos, si es muy grande genera un efecto de atenuación de ruido, pero puede resultar en clases entrecortadas; si es muy pequeño puede generar subajuste. (Elder et.al., 2009)

Es importante mencionar que k-NN es uno de los algoritmos de Machine Learning más utilizados por su simplicidad y precisión, además, es sumamente fácil adaptarlo a nuevas instancias. Por otro lado, es importante tener en cuenta la cantidad y el tipo de datos que se quieren analizar, ya que es sensible al ruido, y al trabajar con demasiados datos puede tener un alto costo computacional, requiriendo de bastante memoria. (USAII, 2023)

2.4 Evaluación del modelo

Al generar un modelo de regresión es necesario medir la precisión de los valores que predice, y esto se lleva a cabo mediante las métricas de evaluación, que están relacionadas con el error. El error se define como la diferencia entre los valores reales y los valores obtenidos mediante el modelo, puede calcularse de distintas maneras, pero las más comunes son MSE, RMSE, MAE y R^2.

2.4.1 Error cuadrático medio

MSE es el tipo de error más simple, aunque los resultados no siempre son informativos o fáciles de interpretar, aun así, es utilizado con frecuencia. Está definido por:

$$MSE = \sum_{i=1}^{N} \frac{\left(y_i - \dot{y}\right)^2}{N},$$

donde y_i Es el resultado real esperado de la i-ésima muestra, $\dot{y}$ Es el resultado obtenido de la i-ésima muestra N Es el número de datos. (Manrique, 2022)

2.4.2 Raíz cuadrada del error cuadrático medio

Un tipo de error similar es RMSE que es, como su nombre lo indica, la raíz cuadrada del MSE.

La ventaja de este tipo de error es que brinda una idea más intuitiva del error obtenido, ya que su magnitud puede ser comparable a las magnitudes que se manejan en el conjunto de datos. Por ejemplo, si los datos están en centímetros y el RMSE es de 0.5, se puede intuir que el error promedio es de 0.5 centímetros.

MSE y RMSE tienen fórmulas similares, y en ambos casos un mejor modelo es el que tiene valores menores. Debido a las similitudes entre ambos, reducir uno de los dos generalmente implica reducir también el otro. (Manrique, 2022)

$$RMSE = \sum_{i=1}^{N} \sqrt{\frac{\left(y_i - \dot{y}\right)^2}{N}}.$$

2.4.4 Normalización

De acuerdo con Google (2025), un tratamiento común a los datos es la normalización, un proceso matemático que consiste en transformar datos numéricos para que se encuentren en una escala definida. Uno de los tipos de normalización más comunes es ajustar los datos a un rango entre 0 y 1. Esto puede lograrse de distintas maneras, entre las que destacan dividir entre el rango y dividir entre el valor máximo. Un ejemplo es NRMSE, que se obtiene dividiendo RMSE entre el rango:

$$NRMSE = \frac{RMSE}{y_{max} - y_{min}}.$$

2.5 Inteligencia artificial en Python

Python es un lenguaje de programación orientado a objetos, que presenta las ventajas de ser libre y de estar ampliamente difundido, por lo que existen una amplia variedad de librerías y documentación que permiten realizar una gran cantidad de tareas, incluyendo aquellas que involucran inteligencia artificial. Las dos librerías más utilizadas en este ámbito son TensorFlow (que utiliza la API Keras para llevar a cabo modelos complejos, como aprendizaje profundo) y ScikitLearn (con módulos que permiten utilizar herramientas simples para el Machine Learning).

ScikitLearn es sumamente útil para trabajar con problemas de regresión y algoritmos, incluyendo k-NN, además de incluir módulos capaces de asistir en el preprocesamiento de los datos, incluyendo conjuntos de prueba, entrenamiento y validación cruzada. Con SciKitLearn es posible generar, ajustar, evaluar, visualizar y predecir instancias a partir de un modelo, por lo que se ha convertido en una herramienta clave en el análisis de datos.

Capítulo 3. Metodología

El estudio se enfocó en analizar datos climatológicos de la Ciudad de Puebla, considerando datos diarios de precipitación, evaporación y temperatura. La evaporación y precipitación se encuentran en milímetros y la temperatura en Celsius. Se buscó generar un conjunto de datos con la mayor cantidad de datos completos posibles para obtener nulos sintéticos. De esta manera, se obtienen dos hiperparámetros para el modelo: k y el peso. A partir de esta información, es posible entrenar el modelo para rellenar nulos reales con el modelo generado.

El proceso se dividió en tres pasos primordiales: adquisición y procesamiento de datos, preprocesamiento y selección del modelo en Python (búsqueda de nulos, aplicación del modelo y cálculos de error en conjunto de prueba para seleccionar valores óptimos), y aplicación del modelo.

3.1 Adquisición y preprocesamiento de datos.

1. Los datos se descargaron de la base de normales climatológicas de la Comisión Nacional del Agua, a través de su página web, se encuentran en formato .txt.
2. Se prepararon los datos en Excel para ingresarlos a Python, eliminando los encabezados y colocando un formato de tabla.

3.2 Procesamiento y selección del modelo

3. El modelo se diseñó en Python utilizando la librería ScikitLearn, siguiendo el flujo a continuación:
    3.1. Transformar las fechas a Julianas, que son números enteros fácilmente manipulables por cualquier algoritmo. Al ser números

consecutivos, se pueden utilizar para obtener un arreglo de fechas completas y consecutivas.

3.2. Ubicando los datos nulos, generar dos conjuntos: uno sin datos nulos y otro donde se incluyan todas las fechas con faltantes.

3.3. Separar un conjunto sin nulos lo más grande posible, para poder extraer “nulos sintéticos”, usando 80% de datos completos y 20% de nulos sintéticos.

3.4. Aplicar el algoritmo k-NN para rellenar los nulos generados de forma sintética.

3.5. Utilizar los valores reales para compararlos con los valores obtenidos utilizando la métrica NRMSE. Buscar el valor de k que minimice el error. Y Calcular métricas de error (RMSE, NRMSE)

3.3 Aplicación del modelo

4. Aplicar el modelo elegido a los nulos reales y calcular los valores faltantes.
5. Ajustar el modelo para las fechas en las que no hay datos.
6. Generar un conjunto con los datos completos.

Resultados

Para llevar a cabo el procesamiento, se utilizó principalmente una computadora personal con 16 GB de memoria RAM y procesador Intel(R) Core(TM) i5-9300HF CPU @ 2.40GHz. Se probaron modelos para distintos valores de k, buscando un valor óptimo para cada atributo; los valores fueron distintos en casi todos los casos.

A continuación se presentan dos tipos de gráficos: En primer lugar, curvas de error. Estas tienen el propósito de buscar el valor óptimo para los hiperparámetros en un algoritmo de Inteligencia Artificial (en este caso k). El código está optimizado

para buscar automáticamente el valor de k con menor error, marcarlo en el gráfico y continuar el procesamiento con dicho hiperparámetro. No obstante (y como ocurrió con la precipitación), es posible modificar el vector con los valores de k en caso de ser necesario. En segundo lugar se muestran los modelos de ajuste, comparando los datos generados por el modelo con los reales.

Se obtuvieron gráficos de curva de error (para buscar el valor que optimice al modelo) y de modelo de ajuste para cada atributo, ya que todos poseen un comportamiento particular.

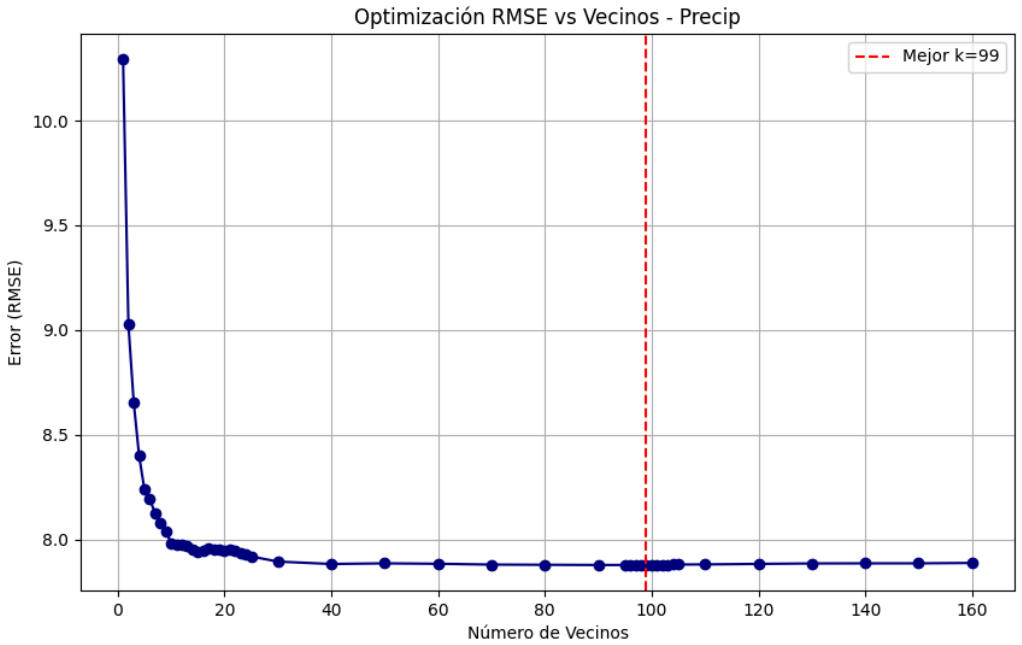


Figura 1: Cálculo de error para distintos valores de k (precipitación).

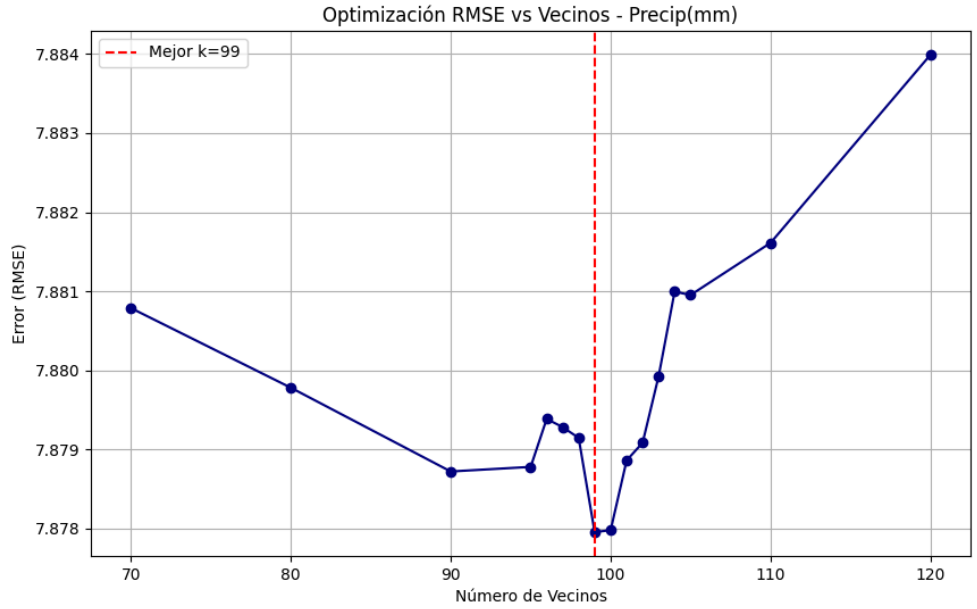


Figura 2: Cálculo de error para distintos valores de k (precipitación), acercamiento

Tal como se observa en la Figura 1, el valor óptimo de k para la precipitación es 99. Esto significa que considerar un número alto de vecinos reduce el error en las predicciones. Esto puede deberse al comportamiento estacional de las lluvias, pero es necesario obtener información adicional para elaborar conclusiones.

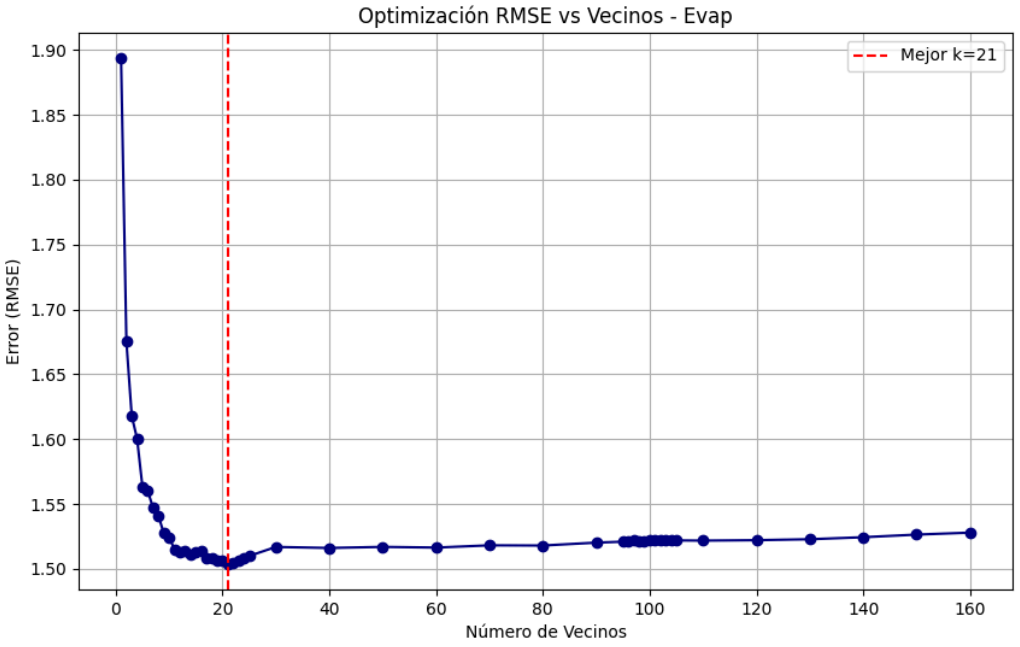


Figura 3: Cálculo de error para distintos valores de k (evaporación).

Según la figura 3, el valor óptimo de vecinos para la evaporación es 21. La evaporación depende, en gran medida, de la temperatura y humedad, por lo que el valor correspondiente a cada día es similar al de los días más próximos.

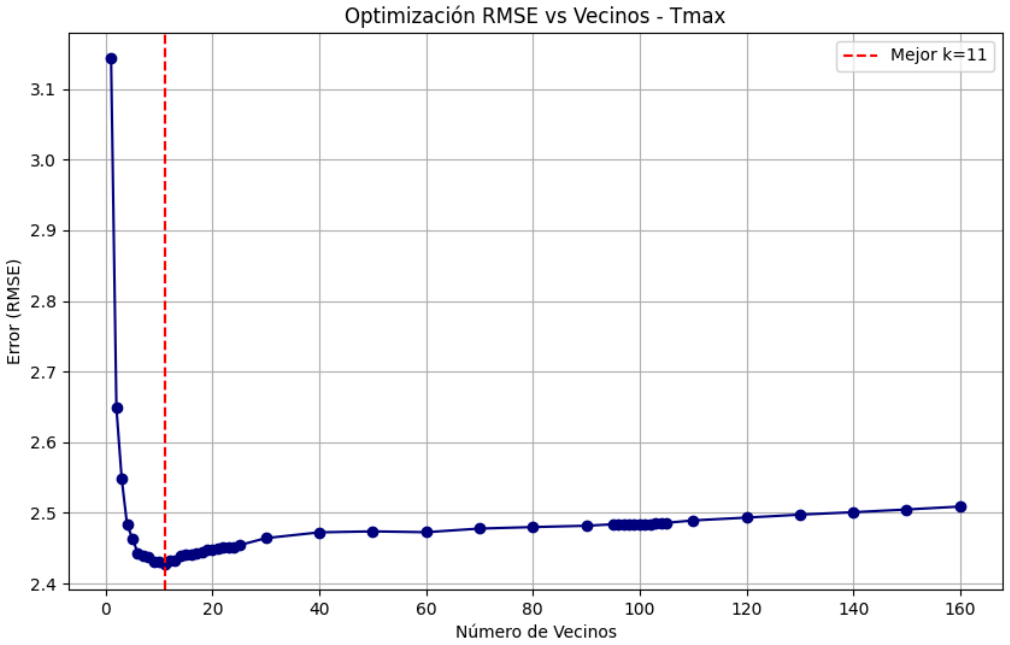


Figura 4: Cálculo de error para distintos valores de k (temperatura máxima).

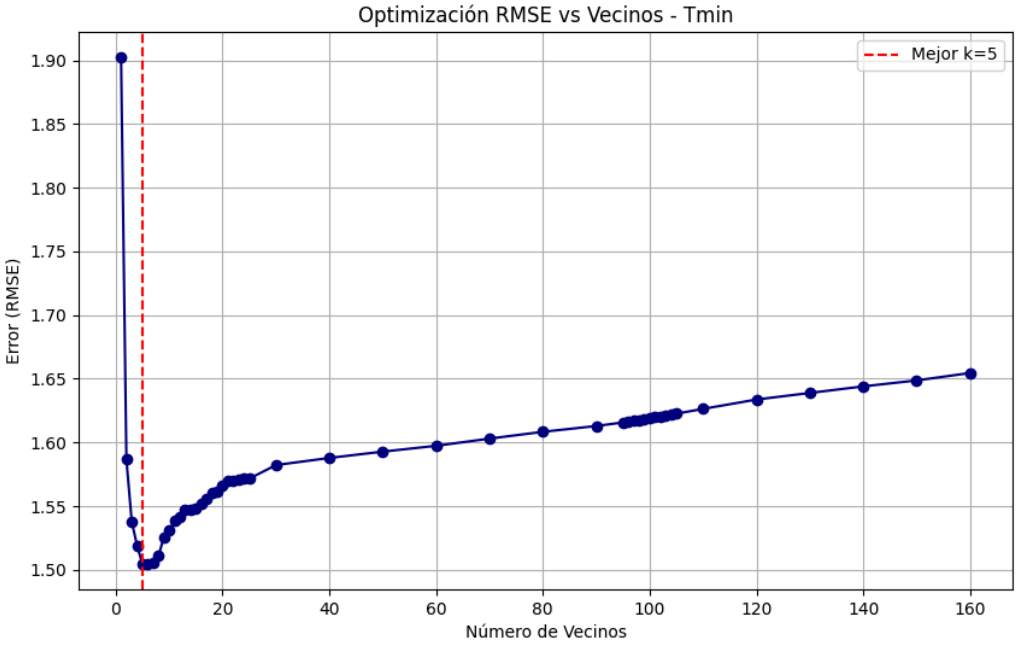


Figura 5: Cálculo de error para distintos valores de k (temperatura máxima).

Como se observa en las figuras 4 y 5, el número ideal de vecinos para la temperatura (máxima y mínima, correspondientemente) es de 11 y 5. Esto tiene sentido considerando que la temperatura no tiende a modificarse drásticamente, y suele ser similar entre días próximos. Es destacable que, al ser atributos de la misma naturaleza (ambos miden la temperatura), el valor óptimo de k es similar.

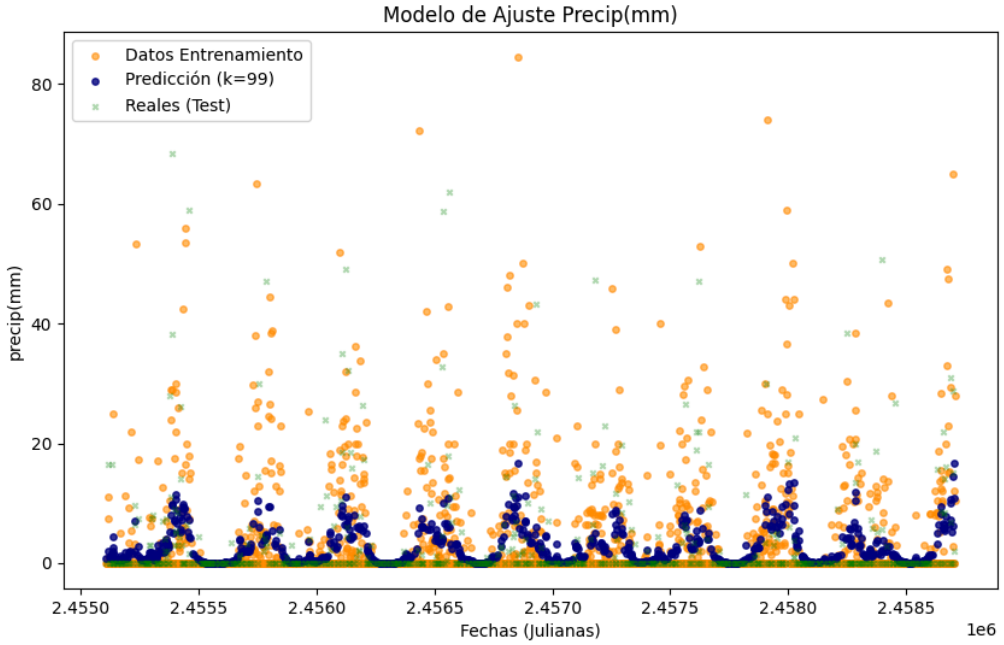


Figura 6: Modelo de ajuste para precipitación

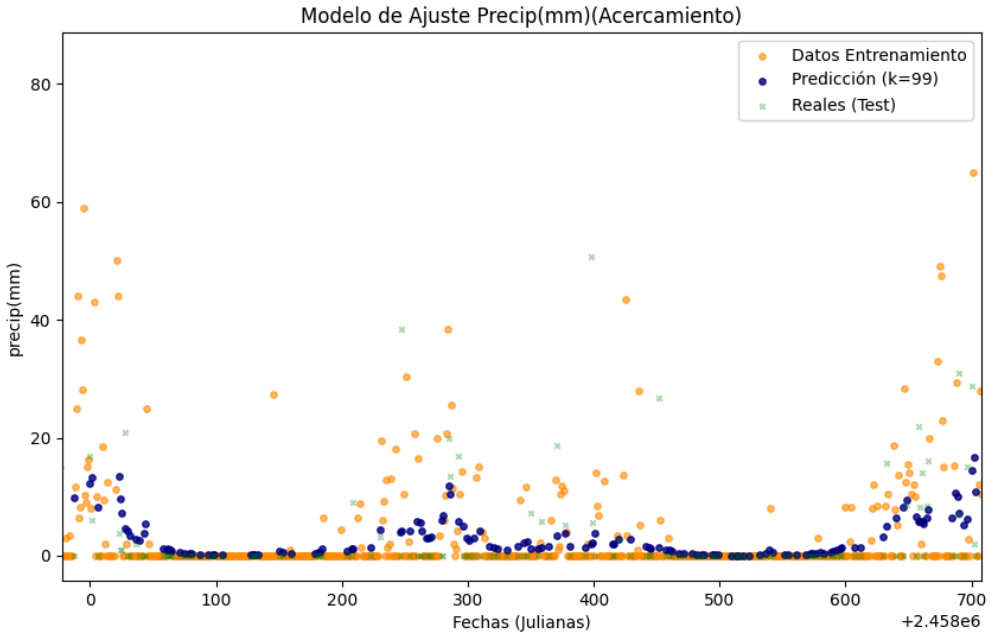


Figura 7: Modelo de ajuste para precipitación (acercamiento)

La figura 6 muestra el modelo de ajuste con k-NN para la precipitación. Es importante destacar que, a pesar de que algunos valores predichos son distintos a los valores reales (especialmente a los que tienden a ser atípicamente altos), el modelo sigue la tendencia de comportamiento de las lluvias a lo largo del periodo analizado. Que el modelo parezca tener un efecto de suavizado y no prediga eficientemente valores altos, puede deberse a que k es un número considerablemente grande. La figura 7 corresponde a un zoom de dos años, entre los que se distingue que, efectivamente, los errores más grandes se encuentran entre los valores excepcionalmente altos.

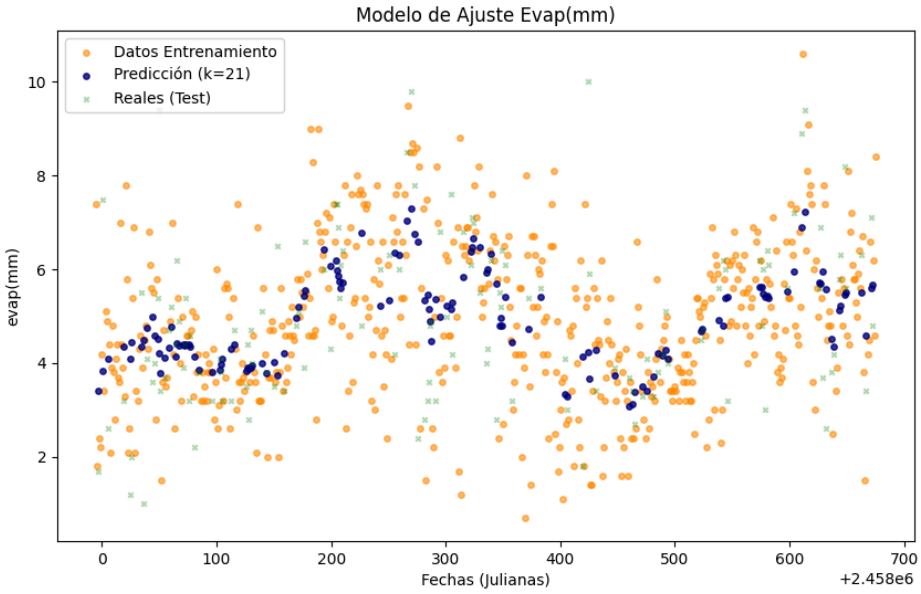


Figura 8: Modelo de ajuste para evaporación.

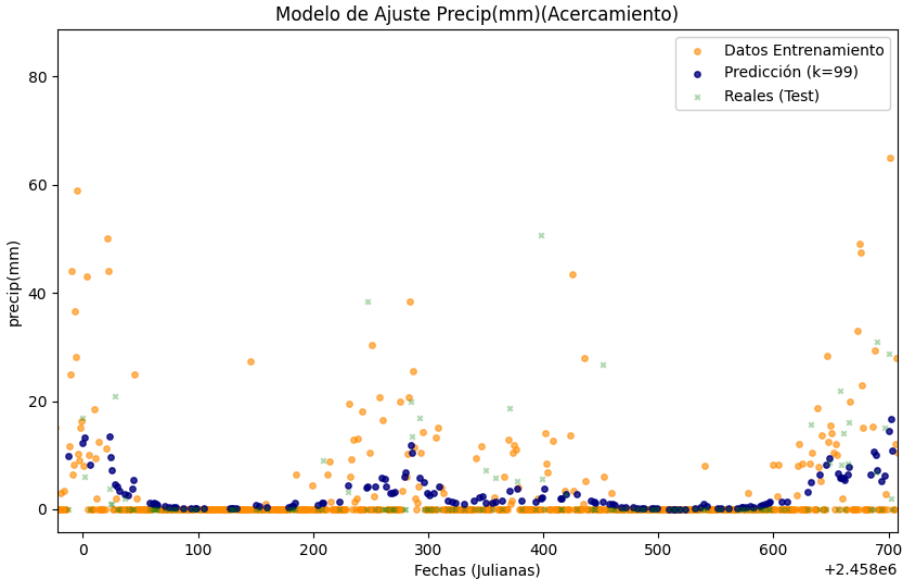


Figura 9: Modelo de ajuste para evaporación (acercamiento a 2 años)

En la figura 8, para el modelo de ajuste para evaporación, podemos observar un comportamiento similar a la precipitación, el modelo de no tiene un desempeño excepcional cuando el faltante es un valor atípico, pero es excelente ajustándose a la tendencia.

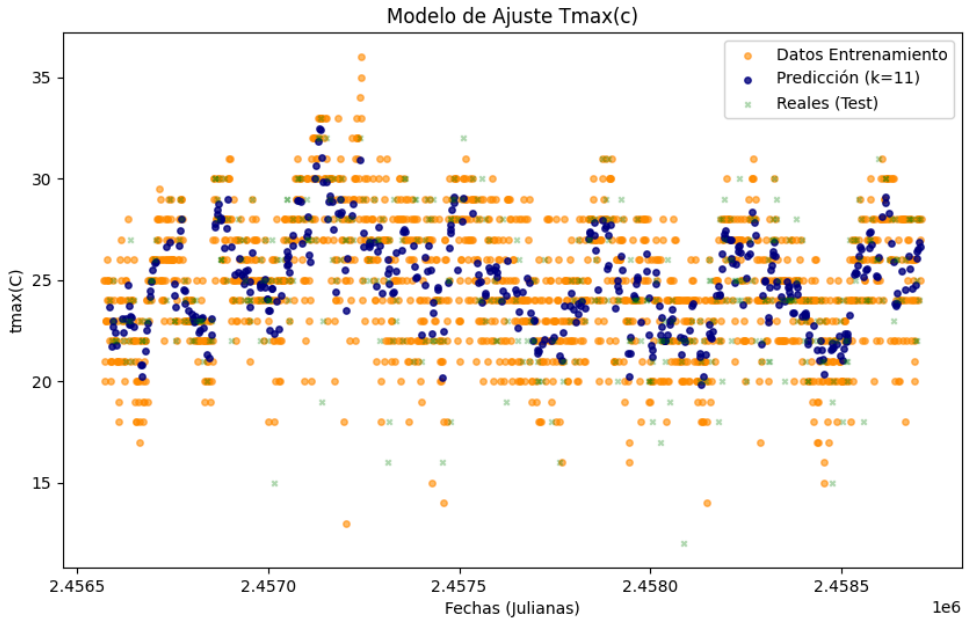


Figura 10: Modelo de ajuste para temperatura máxima

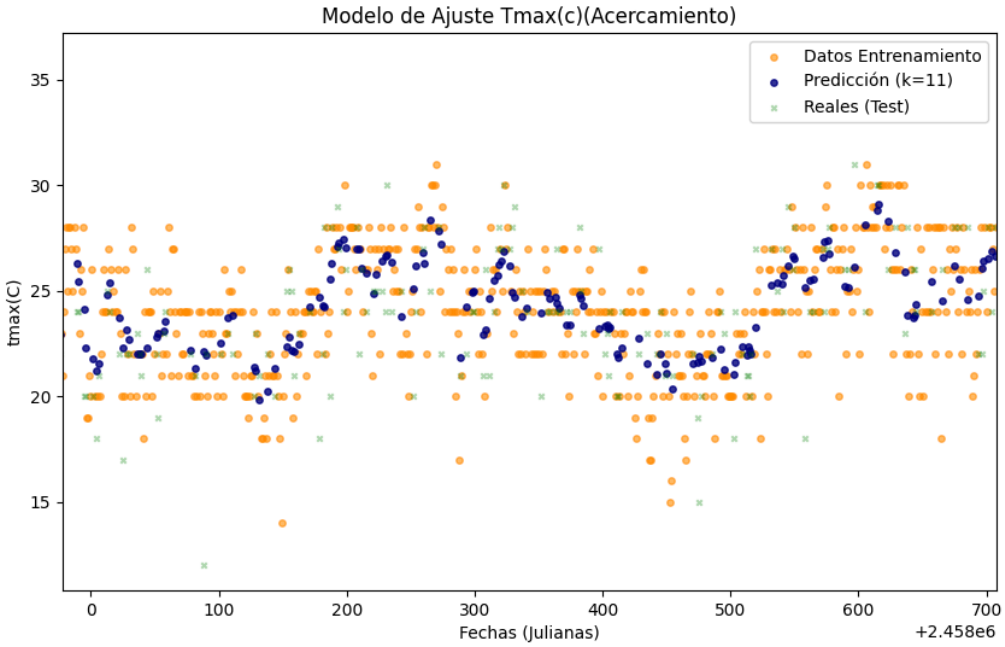


Figura 11: Modelo de ajuste para temperatura máxima (acercamiento a 2 años)

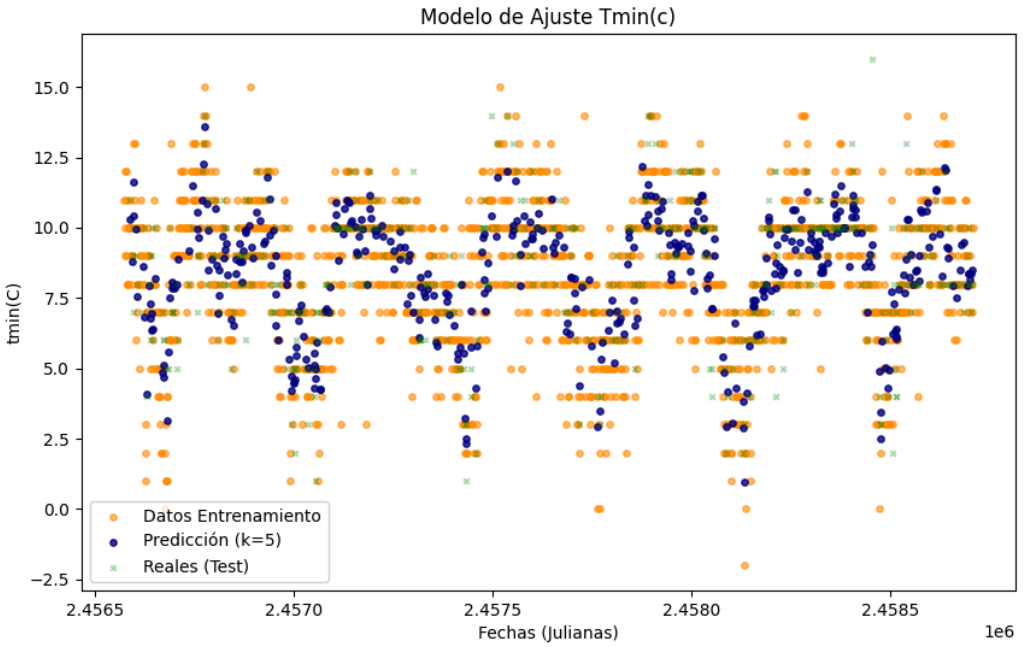


Figura 12: Modelo de ajuste para temperatura mínima

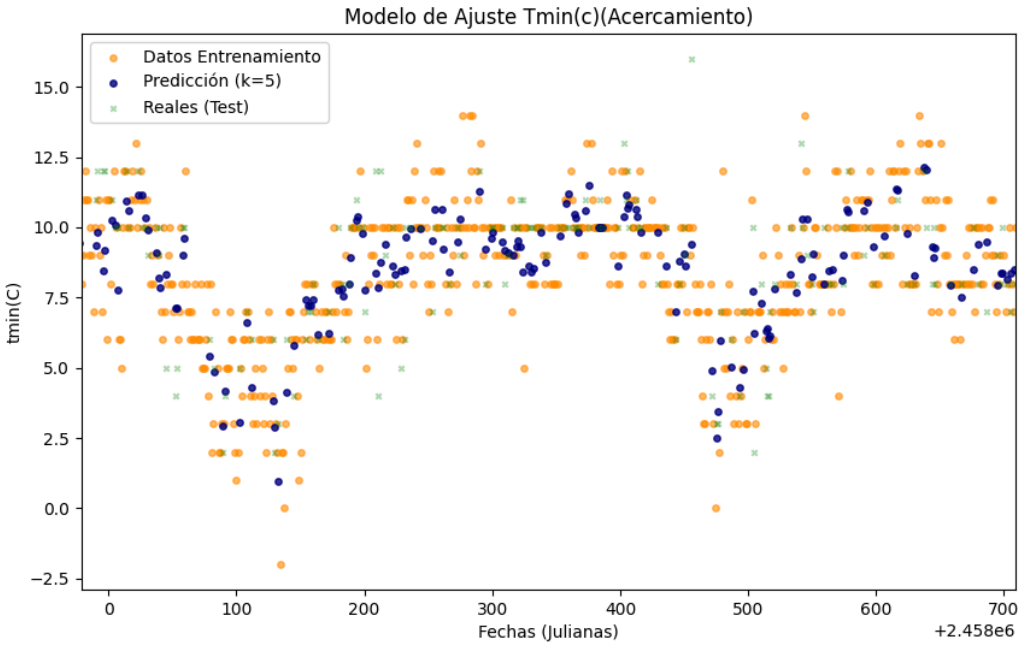


Figura 13: Modelo de ajuste para temperatura mínima

Las figuras 10 y 12 corresponden a los modelos de ajuste para las temperaturas máxima y mínima. El ajuste es bastante bueno, y se observan tendencias similares para ambos casos. Al igual que se observó con las curvas de error, esto es lógico considerando que los atributos corresponden al mismo tipo de medición. En las

figuras 11 y 13 (los acercamientos), se observa que los datos mejor ajustados son aquellos que no se alejan demasiado de la tendencia.

Variable	Mejor K	KNN RMSE	KNN NRMSE	Media RMSE	Media NRMSE	Median a RMSE	Mediana NRMSE	Moda RMSE	Moda NRMSE
precip(mm)	99	7.877	0.115	8.448	0.123	8.998	0.131	8.998	0.131
evap(mm)	21	1.503	0.167	1.7565	0.195	1.759	0.195	1.782	0.198
tmax(C)	11	2.427	0.115	3.276	0.156	3.292	0.156	3.336	0.158
tmin(C)	5	1.504	0.1003	2.443	0.162	2.485	0.165	2.850	0.190

Tabla 1: Cálculos de error con distintas métricas (k-NN vs medidas de tendencia central)

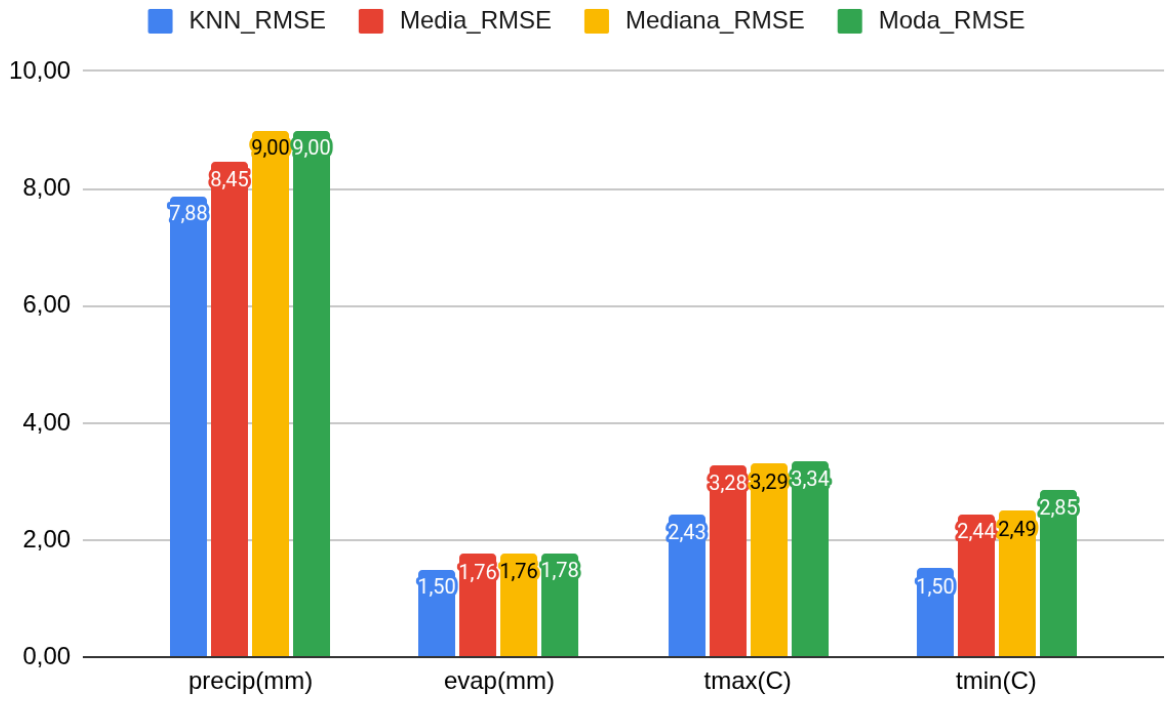


Figura 14: Comparativa de error RMSE para imputación con k-NN y medidas de tendencia central.

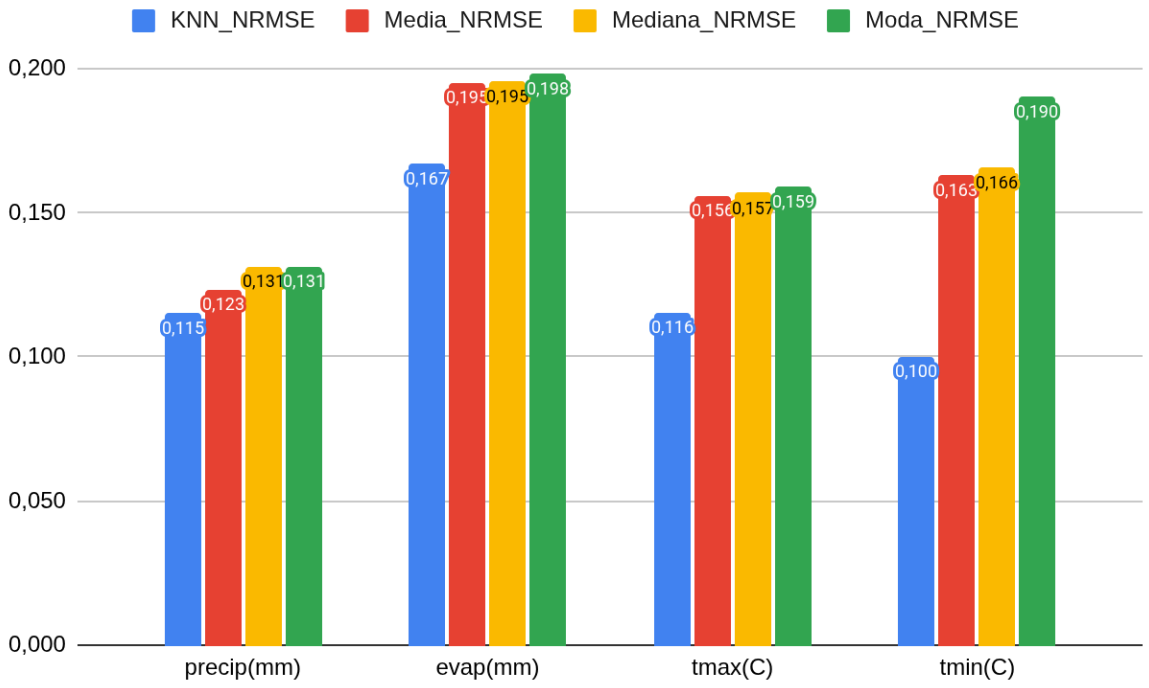


Figura 15: Comparativa de error NRMSE para imputación con k-NN y medidas de tendencia central.

La tabla 1 contiene los cálculos para cada medida empleando RMSE y NRMSE. En cuanto a las figuras 14 y 15, funcionan como comparativa visual de ambos errores para cada medida y método de imputación. Tanto en la tabla como en los gráficos de barras, es notable que los errores obtenidos con k-NN son menores que los correspondientes a las medidas de tendencia central. El mejor ajuste para el modelo corresponde a ambas temperaturas. Por el contrario, la precipitación tuvo los errores más grandes.

Es importante destacar que el algoritmo posee una gran capacidad para identificar una tendencia en un conjunto. Además, en múltiples pruebas realizadas, el tiempo de ejecución jamás superó los 20 segundos; de manera que no se requiere una gran inversión de tiempo máquina.

Asimismo, y según afirman Valencia et.al. (2022), para estaciones con menos de 50 km de distancia horizontal y 750 m de altura, el comportamiento climático es

similar, por lo que es de esperarse que este modelo sea fácilmente aplicable y proporcione resultados similares para otras estaciones del área urbana.

Conclusiones

El algoritmo k-NN demostró ser una herramienta valiosa en el rellenado de datos periódicos, con una tendencia marcada, o donde los datos se encuentren influenciados por sus vecinos. El código generado puede utilizarse satisfactoriamente para rellenar datos nulos, con poca intervención del usuario, y una gran facilidad para reemplazar por otros conjuntos de datos. Además, se consiguió mejorar el error sin aumentar demasiado el tiempo y los recursos de cómputo, alcanzando equilibrio rapidez, practicidad y precisión.

Los valores óptimos de k no son iguales para todas las métricas, esto tiene sentido considerando que no todas las métricas climáticas se comportan de la misma manera. Se pueden observar tendencias marcadas en la temperatura, pero la precipitación y evaporación son mucho más variables. Esto puede deberse a que la temperatura tiene una mayor correlación con la época del año, mientras que las otras dos dependen de otros factores, y se comportan de manera variable de un año a otro.

Este proyecto demuestra que la aplicación de algoritmos de Inteligencia Artificial a las Ciencias de la Tierra es, no sólo posible, sino esencial para optimizar procesos y obtener resultados más precisos. En el caso particular de k-NN, se perfila como una herramienta con enorme potencial para distintos tipos de variables y métricas. Por ejemplo, los registros de pozo, o de SEV (Sondeos Eléctricos Verticales) con frecuencia presentan valores nulos, y sus señales llegan a comportarse de forma periódica o con tendencias dependientes de sus vecinos. El

tratamiento de nulos para estos casos usualmente elimina los faltantes o reemplaza con números sin significado físico. k-NN, presenta una alternativa eficaz para este tipo de tratamientos, por lo que es importante explorar sus aplicaciones en estos casos.

Sondear este tipo de análisis es imprescindible para la Ingeniería Geofísica, siendo prometedores para mejorar la calidad y rapidez con que se realizan los análisis en esta área. Además, las Ciencias de la Tierra se benefician enormemente del software especializado, pero los académicos en el campo pocas veces estudian la implementación de nuevas herramientas tecnológicas en este ámbito. Así, la bibliografía al respecto es reducida, especialmente en español, desaprovechando herramientas valiosas, no solo en el ámbito técnico y productivo, sino también en la investigación.

Por lo tanto, explorar nuevas aplicaciones de tecnologías emergentes es fundamental para obtener avances y soluciones a problemas, tanto existentes como emergentes, a los que constantemente se enfrentan estas ciencias (cambio climático, análisis de riesgos, sondeos geofísicos, etc.); además de proporcionar oportunidades para comprender, desde distintos puntos de vista, los fenómenos que se estudian en esta área.

Referencias

Amazon Web Services. (2017) “IA Y Machine Learning: Diferencia Entre Inteligencia Artificial Y Machine Learning. AWS.” *Amazon Web Services,* aws.amazon.com/es/compare/the-difference-between-artificial-intelligence-and-machine-learning/.

Comisión Nacional del Agua. (2024). Normales Climatológicas por Estado. Smn.conagua.gob.mx. https://smn.conagua.gob.mx/es/informacion-climatologica-por-estado?estado=pue

Davi, T. (2023) "Understanding Different Distance Measures"-. LinkedIn https://www.linkedin.com/pulse/understanding-different-distance-measures-tiago-davi-1f

Duarte, Luíza & Formiga, Klebber & Costa, Veber. (2022). Comparison of Methods for Filling Daily and Monthly Rainfall Missing Data: Statistical Models or Imputation of Satellite Retrievals?. Water. 14. 3144. 10.3390/w14193144.

Elder, J., Miner, G., Nisbet, R. (2009) Classification. *Handbook of Statistical Analysis and Data Mining Applications* (pp. 235-258). Elsevier. https://doi.org/10.1016/b978-0-12-374765-5.00011-5. https://doi.org/10.1016/b978-0-12-374765-5.00011-5.

Géron, A. (2019). Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: concepts, tools, and techniques to build intelligent systems (2nd ed.). O'Reilly.

Google. (2025). Datos numéricos: Normalización. Google for Developers. https://developers.google.com/machine-learning/crash-course/numerical-data/normalization?hl=es-419

Alemán-Rivas, Y., Cabrera-Hernández, L., Casas-Cardoso, G., Chávez-Cárdenas, M., Díaz-Barrios, H., Morales-Hernández, A. (2015) "Algoritmos de aprendizaje automático para clasificación de Splice Sites en secuencias genómicas". SciELO - Scientific Electronic Library Online. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992015000400012

IBM (2022) "What is the k-nearest neighbors algorithm?" IBM. https://www.ibm.com/topics/knn

IBM. (2020). ¿Qué es la inteligencia artificial (IA)? IBM. https://www.ibm.com/mx-es/topics/artificial-intelligence

Imam, F., Musilek, P. y Reformat, MZ (2024). Técnicas de aprendizaje automático paramétrico y no paramétrico para aumentar la confiabilidad de los sistemas de potencia: Una revisión. Información , 15 (1), 37. https://doi.org/10.3390/info15010037

J. Sun, W. Du y N. Shi (2018) "A Survey of kNN Algorithm", Inf. Eng. Appl. Comput. https://www.researchgate.net/publication/348305327_A_Survey_of_kNN_Algorithm

McHugh M. L. (2013). The chi-square test of independence. Biochemia medica, 23(2), 143–149. https://doi.org/10.11613/bm.2013.018

Park, J., Müller, J., Arora, B. et al. Long-term missing value imputation for time series data using deep neural networks. Neural Comput & Applic 35, 9071–9091 (2023). https://doi.org/10.1007/s00521-022-08165-6

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

Schneider, P., & Xhafa, F. (2022). Anomaly detection. In Anomaly Detection and Complex Event Processing over IoT Data Streams (pp. 49–66). Academic Press. https://doi.org/10.1016/b978-0-12-823818-9.00013-4

SciKit-Learn (2025) R2 score, the coefficient of determination. https://scikit-learn.org/stable/modules/model_evaluation.html#r2-score

SciKit-Learn. (2011). sklearn.neighbors.KNeighborsRegressor. Scikit-Learn. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html#sklearn.neighbors.KNeighborsRegressor.score

United States Artificial Intelligence Institute (2023) “Understanding KNN Algorithm and Its Role in Machine Learning”. https://www.usaii.org/ai-insights/understanding-knn-algorithm-and-its-role-in-machine-learning.

Valencia Gonzalez, J. N., Ramírez, R. A., Peña, M. A. V., & Quevedo Nolasco, A. (2022). Relleno de datos diarios faltantes en registros de series climatológicas temporales. Revista Mexicana De Ciencias Agrícolas, 13(4), 617–629. https://doi.org/10.29312/remexca.v13i4.2514

Zas Pérez, Alexandre (2023) Técnicas de tratamiento de datos faltantes y aplicación en problema de detección de fraude bancario, Repositorio Institucional Universidade Da Coruña.