

TECNOLÓGICO NACIONAL DE MÉXICO, CAMPUS CULIACÁN

DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN



**RECONSTRUCCIÓN DE LA INFORMACIÓN DE PARÁMETROS
DE LA CALIDAD DEL AGUA DE SINALOA, MEDIANTE
TÉCNICAS DE APRENDIZAJE MÁQUINA**

Tesis para obtener el título de Ingeniero en Sistemas Computacionales

MEDRANO ITURRIOS ANGEL RAÚL

ASESORES

DR. HECTOR RODRIGUEZ RANGEL

MCC. GLORIA EKATERINE PERALTA PEÑUÑURI

Culiacán, Sinaloa

Noviembre 2024



DIVISION DE ESTUDIOS PROFESIONALES
ASUNTO: **Autorización de Impresión**

Culiacán, Sinaloa, 16/10/2023 16:47

C. ANGEL RAUL MEDRANO ITURRIOS
PASANTE(S) DE LA CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES
P R E S E N T E.

De acuerdo con fallo emitido por la Comisión Revisora Integrada por:

Cédula
Profesional

Presidente: Doctorado en Ciencias en Ingeniería Eléctrica Opción en
Sistemas Computacionales

Héctor Rodríguez Rangel

9972835

Secretario: Maestría en Ciencias en Ciencias de la Computación

Gloria Ekaterine Peralta Peñuñuri

6038275

Vocal: Doctorado en Ciencias de la Computación

María Lucia Barrón Estrada

8630269

Y considerando que reúne los requisitos establecidos en la Opción: **TITULACIÓN INTEGRAL: TESIS PROFESIONAL "RECONSTRUCCIÓN DE LA INFORMACIÓN DE PARÁMETROS DE LA CALIDAD DEL AGUA DE SINALOA, MEDIANTE TÉCNICAS DE APRENDIZAJE MÁQUINA"**, damos a usted nuestra autorización para que proceda a imprimir su trabajo profesional.

Sin otro particular, quedo de usted.

ATENTAMENTE

Excelencia en Educación Tecnológica®

CONCEPCION MENDOZA ROSALES
LIC. CONCEPCION MENDOZA ROSALES
JEFA DE LA DIVISION DE ESTUDIOS PROFESIONALES



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE CULIACÁN

DEPARTAMENTO DE DIVISIÓN DE
ESTUDIOS PROFESIONALES



Índice general

1	INTRODUCCIÓN	1
1.1	Definición del problema	6
1.2	Hipótesis	7
1.3	Objetivos	8
1.3.1	Objetivo General	8
1.3.2	Objetivos Específicos	8
1.4	Justificación	8
1.5	Estructura de la tesis	10
2	MARCO TEÓRICO	11
2.1	Inteligencia artificial (Artificial Intelligence/AI)	11
2.1.1	Aprendizaje Máquina (Machine Learning/ML)	13
2.1.2	Aprendizaje Supervisado (Supervised Learning)	14
2.1.3	Aprendizaje no Supervisado (Unsupervised Learning)	15
2.2	Redes Neuronales (Neural Networks)	17
2.2.1	Red Neuronal Artificial (Artificial Neural Network/ANN)	17
2.2.2	Red Neuronal Convolucional (Convolutional Neural Network/CNN)	20
2.3	Métricas de evaluación	22
2.3.1	R-Cuadrado (Coeficiente de Determinación)	22
2.3.2	Error Cuadrático Medio (Mean Squared Error/MSE)	25
2.3.3	Desviación Cuadrática Media (Root Mean Squared Error/RMSE)	26
2.4	Datos Faltantes (Missing Data)	27

2.4.1	Mecanismos de perdidas	27
2.4.2	Datos Faltantes Completamente Aleatorios (Missing Data Completely At Random/MCAR)	28
2.4.3	Datos Faltantes Aleatorios (Missing Data At Random/MRA)	29
2.4.4	Datos Faltantes No Aleatorios (Missing Data Not Random/MNRA)	29
2.5	Técnicas y métodos para el tratamiento de datos faltantes	30
2.5.1	Eliminación Por Lista (List-wise Deletion)	30
2.5.2	Imputación Simple	31
2.5.3	Regresión Lineal	31
2.5.4	Imputación Múltiple	32
2.5.5	Algoritmo K Vecinos Más Cercanos (K-Nearest Neighbor/KNN)	33
2.6	Tecnologías y recursos	34
2.6.1	Python	34
2.6.2	scikit-learn	38
2.6.3	Matplotlib	39
2.6.4	Pandas	43
2.6.5	NumPy	44
2.6.6	TensorFlow	45
3	ESTADO DEL ARTE	46
3.1	Trabajos relacionados	46
4	METODOLOGÍA	48
4.1	Proceso de limpieza y filtración	50
4.1.1	Panorama general del conjunto de datos	50
4.1.2	Área de interés	52
4.1.3	Proceso de filtración	53
4.1.4	Proceso de limpieza	57
4.2	Proceso de exploración y análisis de los datos	59

4.2.1	Identificando mecanismos de perdidas	62
4.3	Proceso de imputación	66
4.3.1	Análisis de caso de estudios completo	67
4.4	Proceso de generación de modelos de aprendizaje máquina	72
5	RESULTADOS	75
5.1	Resultados obtenidos a través de los métodos de imputación	75
5.2	Resultados obtenidos a través de los modelos de aprendizaje máquina . .	79
5.2.1	Resultados obtenidos a través de CNN	79
5.2.2	Resultados obtenidos a través de ANN	82
5.2.3	Resultados al agregar más parámetros	84
6	CONCLUSIÓN	87
6.1	Recomendaciones	88
	REFERENCIAS	89
	ANEXOS	93
A	Tabla de parámetros	93
B	Matriz de correlación entre parámetros	95

Índice de figuras

2.1	Test de Turing	12
2.2	Flujo de trabajo de un modelo de ML con aprendizaje supervisado	14
2.3	Aprendizaje No Supervisado	16
2.4	Estructura de una ANN	18
2.5	ANN: Caja negra (Zupan, 1994)	19
2.6	Patrón de tipo cuadrado (Yamashita et al., 2018)	20
2.7	Grados de correlación lineal (Jierula et al., 2021)	23
2.8	Eliminación por lista	30
2.9	Algoritmo KNN	33
2.10	Versiones Python	34
2.11	Dependencias de scikit-learn	38
2.12	Gráfica de la función seno con Matplotlib	42
4.1	Panorama general de la base de datos	52
4.2	Tabla sitios	53
4.3	Filtro para Sinaloa	54
4.4	Sitios filtrados para Sinaloa	54
4.5	Filtro claves Sinaloa	55
4.6	Monitoreos de Sinaloa	56
4.7	Macro de VB para eliminar Subterráneo	56
4.8	Registros con carácter “<”	57
4.9	Función buscar y reemplazar	58

4.10	Valores faltantes	59
4.11	Valores repetitivos	61
4.12	Mecanismos de perdidas	62
4.13	Matriz de correlación de perdida	63
4.14	Presencia de valores MAR	64
4.15	Metodología para generar modelos de imputación	68
4.16	Distribución de los datos	71
4.17	Distribución de los datos despues de imputar	71
4.18	Comparación de la distribución de los datos	71
4.19	Metodología para generar modelos de aprendizaje máquina	72
5.1	Resultados obtenidos para N_NO3 con CNN	80
5.2	Valores faltantes con porcentajes de hasta 50 %	84

Índice de tablas

3.1	Estado del Arte sobre Manejo de Datos Faltantes	47
4.1	Valores faltantes	60
4.2	Mecanismos de pérdidas	65
5.1	Resultados al imputar COLI_FEC	76
5.2	Resultados al imputar COT y COT_SOL	77
5.3	Resultados al imputar N_NH3, N_TOT, N_TOTK, P_TOT, ORTO_PO4 .	77
5.4	Resultados al imputar N_NON2	77
5.5	Resultados al imputar N_ORG	78
5.6	Resultados obtenidos CNN (10)	79
5.7	Resultados obtenidos para N_NO3 con CNN	80
5.8	Resultados obtenidos para SDT, COLOR_VER, ABS_UV, SST, TURBIEDAD y TEMP_AMB con CNN	81
5.9	Resultados obtenidos con ANN 1	82
5.10	Resultados obtenidos con ANN 2	83
5.11	Resultados finales obtenidos con CNN	85
5.12	Resultados finales obtenidos con ANN	86

Dedicatoria

A mis padres, **Alma** y **Raúl**, les dedico con mucho amor, aprecio y cariño este trabajo por ser ustedes los principales merecedores de todos los éxitos que su hijo pueda obtener.

A mis hermanas, principalmente a **Yuniria** por acompañarme en este trayecto y ser mi compañera de noche y de día. Y así mismo, a mi hermana **Arleth** para que siempre vea a su hermano como ejemplo y motivación. De la misma manera, a mi hermano mayor **Adid** por acompañarme en mis primeros años de universidad.

A mi **familia** y **amigos más cercanos**, todo esto es por ustedes y para ustedes.

Agradecimientos

*Con cariño y desde lo más profundo de mi corazón, para cada uno de
ustedes.*

-Angel Raúl Medrano Iturrios.

Al **Tecnológico Nacional de México, Campus Culiacán** por brindarme una educación de calidad, porque aquí es donde me desarrollé y me formé no solo como un profesionista, sino también como una persona de bien y aprendí tanto como de educación y sobre la vida misma. Por todas las aventuras y bonitos momentos que pasé aquí en la universidad, gracias por ser parte de mi vida y mi destino. De la misma manera, por proporcionarme las herramientas y habilidades necesarias para afrontar con éxito los próximos retos en mi vida laboral. Por haberme reunido con excelentes personas que se convirtieron en grandes amistades y que solo aquí pudiese haber encontrado. Te estaré eternamente agradecido y siempre me sentiré orgulloso de haber formado parte de esta linda institución, no es un adiós, es un hasta que la vida nos vuelva a reencontrar, gracias por todo.

Así mismo, agradezco a **cada uno de mis profesores** por ser ustedes principalmente quienes me compartieron de sus conocimientos y también por su entrega, esfuerzo y compromiso ejercido para que yo pudiese aprender de ustedes. Por sacar lo mejor de mí, por ser parte de mi desarrollo profesional y de este bonito proceso de la vida llamado universidad. Estoy en deuda con todos ustedes y les estaré eternamente agradecido por todo lo que un día me enseñaron. De la misma manera les digo que tienen un lugar en mi corazón, en especial para la maestra **Judith B.** y para el maestro **V. Bátiz**. Me llevo lo mejor de cada uno de ustedes y les digo gracias por haber sido mis maestros, con respeto y mucha admiración.

A mis asesores, a la **maestra Katty** por ser una excelente docente, por esa única y peculiar manera de compartir sus conocimientos. De igual manera por siempre fomentar y acercar a los alumnos a estos temas de investigación, gracias por darme la oportunidad de esta nueva experiencia y por todo lo que me enseñó durante sus clases. También al **Dr. H. Rangel** por ser un excelente docente y asesor, así mismo por tener siempre la mejor disposición para apoyarme atendiendo todas mis dudas y aconsejándome de la mejor manera con su grata experiencia. De la misma manera agradezco al **maestro J.L. Medina** por permitirme colaborar en este proyecto, por todos sus consejos, por compartir de su conocimiento conmigo y así mismo por siempre tener la mejor disposición para atender siempre todas mis dudas e inquietudes. Gracias por su confianza, por ser parte de mi desarrollo profesional y por todo su apoyo, ya que sin el esto no hubiese sido posible.

Así mismo, me permito externar mis agradecimientos para el **Departamento de la División de Estudios de Posgrado e Investigación del Tecnológico Nacional de México, Campus Culiacán** por aceptarme y darme la oportunidad de fungir como residente durante el periodo enero-junio 2023.

A **mis compañeros de generación**, gracias por ser parte de esta bonita etapa de la vida llamada universidad y por hacer más amena mi estancia aquí. Porque con ustedes aprendí el valor del compañerismo, compartí risas y carcajadas, así mismo muchos momentos de felicidad y otros tantos un poco amargos y difíciles, pero fue gracias a su apoyo que logré salir adelante. También por haberme compartido de su conocimiento, siempre les estaré eternamente agradecido y tienen un lugar en mi corazón, en especial para mis compañeros **Rogelio, Angel y César**. No les digo adiós, ya que nuestra amistad no termina aquí y además, estoy seguro de que el destino se encargará de reencontrarnos muy pronto.

A **mis padres** por haberme dado la mejor herencia de todas, la educación. Por su inigualable e incomparable amor, por todo su incondicional apoyo y porque ustedes son mi fuente de inspiración y motivación. Porque cuando sentía no poder más fueron ustedes precisamente quiénes me dieron la fuerza para seguir adelante. Por una vida llena de sacrificios que no me alcanzarían las palabras para describirles lo agradecido que estoy con ustedes, pero quiero decirles que aquella semilla que con tanto amor sembraron juntos un día en mi corazón, hoy comienza a dar frutos y quiero que sientan que el objetivo logrado también es de ustedes, ya que los éxitos son suyos y yo solo me encargo de hacérselos llegar. Gracias, **mamá y papá**, con cariño y admiración, su hijo.

A mi **hermana Yuniria**, por ser tu mi compañía de noche y de día, por acompañarme en este viaje, por estar conmigo y por no dejarme solo. Porque fuiste tu mi consuelo después de un largo día, porque me demostraste y enseñaste el amor entre hermanos y porque siempre me apoyaste, así mismo por ser parte de mi inspiración y motivación. No existen las palabras para agradecer todo lo que hiciste por mí, pero quiero decirte que siempre estaré para ti y que espero con ansias que llegue el día en que te encuentres a pasos de culminar tus estudios profesionales para poder regresarte el favor y que puedas así entender el significado de estas palabras, con cariño y aprecio, tu hermano.

A **Don Charly** y a su esposa **Rosy**, por brindarnos un lugar seguro y tranquilo a mi hermana y a mí, por hacernos sentir como en casa y por siempre estar al pendiente de nosotros, así mismo por todas sus muestras de cariño y porque ustedes también son parte de este proceso. No hay palabras suficientes para agradecer todo lo que han hecho por nosotros, pero quiero decirles que tienen un lugar especial en mi corazón, que siempre los voy a recordar y que estoy agradecido con Dios por haberlos puesto en mi camino, gracias por todo su apoyo, que Dios los bendiga siempre, con cariño y aprecio.

A mis **amigos más cercanos**, por ser ustedes quiénes me apoyaron en momentos difíciles y de angustia, por sus muestras de cariño y apoyo hacía mi persona, por los miles de aventuras que juntos hemos compartido, gracias por su bonita y verdadera amistad.

A mi gran amigo y compañero, **Arturo Sabinas Figueroa** (1996-2021) por su bonita y desinteresada amistad, por haber sido un excelente mentor, pero sobre todo un gran amigo. Estaré eternamente agradecido contigo y te llevaré siempre en mi corazón, junto con todos aquellos bonitos momentos que un día me regalaste.

A **mí mismo**, porque solo yo sé lo que me ha costado el llegar hasta aquí, porque soy yo el arquitecto de mi propio destino y a pesar de todas las adversidades, hemos logrado cumplir con el objetivo que un día nos propusimos. Gracias por no desistir.

Y por último, pero no menos importante, a **Dios** y a la vida misma, por tenerme aquí y por este camino recorrido. Por mostrarme que puedo superarme día con día y por enseñarme que los días malos al final de cuentas no son tan malos, ya que gracias a esto es como he podido ir creciendo y mejorando personalmente. También a todas aquellas personas que de una forma u otra aportaron su granito de arena creyendo en mí y que me supieron brindar muestras de apoyo, no me queda más que decirles muchas gracias a todos.

Resumen

Dado que los datos faltantes son un problema recurrente en todo sistema informático y que no se puede estar exento de este suceso, es indispensable contar con estrategias que permitan mitigar y dar un tratamiento efectivo a estos mismos. La ausencia de valores afecta de manera directa en todos los análisis de datos y particularmente cuando se desea trabajar con modelos de aprendizaje máquina, puesto que la presencia de estos entorpece dicho proceso. Tal es el caso de este estudio, donde se tiene una fuente de información en el cual se albergan las muestras recabadas a través de los distintos cuerpos de agua en el estado de Sinaloa desde el año 2012 hasta 2020 esto con la finalidad de determinar la calidad de esta sustancia, más sin embargo la presencia de las grandes cantidades de datos faltantes que se tiene ha frenado a los expertos del área cuando de realizar análisis se trata. Por eso una solución planteada fue realizar una reconstrucción de estos utilizando técnicas estadísticas y de aprendizaje máquina, las distintas estrategias como lo son imputación simple y múltiple ayudan con este proceso, más sin embargo no son del todo precisas, en especial la primera. Dicha estrategia es útil cuando las variables tienen valores muy repetitivos, puesto que estas técnicas generalmente remplazan los valores faltantes con otros determinados a partir de medias de tendencias centrales como lo son media o moda, por lo que el dicho valor suele ser el mismo. En cambio, la imputación múltiple trabaja de manera iterativa y es útil con datos que tienen una correlación lineal. Así también, la imputación va de la mano con la predicción a través de modelos de aprendizaje máquina ya que es necesario realizar un proceso de imputación

para poder utilizar dichos modelos. Unas de las redes que mejores resultados brindaron fueron las **ANN** las cuales resultan útiles para aquellos valores donde no se tiene una buena correlación lineal. También se emplearon redes **CNN**, mismas que revelaban resultados muy cercanas a las anteriores. Después de poner en marcha las estrategias de imputación se procedió a realizar la generación de modelos de aprendizaje máquina los cuales superaron notablemente a dichas estrategias, de las 256 variables que se tenían fue posible reconstruir un total de 23, mismas que fueron seleccionadas y analizadas para determinar si era factible reconstruirlas, dichas variables contenían valores de perdidas desde un 0 % hasta un 50 %.

Abstract

Given that missing data is a recurring problem in every computer system and that one cannot be exempt from this event, it is essential to have strategies to mitigate and effectively treat them. The absence of values directly affects all data analysis and particularly when you want to work with machine learning models, the presence of these hinders said process. Such is the case of this study, where there is a source of information in which the samples collected through the different bodies of water in the state of Sinaloa are housed from 2012 to 2020 with the purpose of determining the quality of this substance, however the presence of the large amounts of missing data that we have has slowed down experts in the area when it comes to carrying out analysis. That is why one solution proposed was to reconstruct them using statistical and machine learning techniques. Different strategies such as simple and multiple imputation help with this process, however, they are not entirely accurate, especially the first. This strategy is useful when the variables have very repetitive values, since these techniques generally replace the missing values with others determined from the mean or mode, so the said value is usually the same. Instead, multiple imputation works iteratively and is useful with data that has a linear correlation. Likewise, imputation goes hand in hand with prediction through machine learning models since it is necessary to carry out an imputation process to be able to use these models. Some of the networks that provided the best results were the ANNs, which are useful for those values where there is no good linear correlation. CNN networks were also used, which revealed results very close to the previous ones.

After implementing the imputation strategies, the generation of machine learning models was carried out, which significantly outperformed these strategies. Of the 256 variables that were available, it was possible to reconstruct a total of 23, which were selected and analyzed to determine if it was feasible to reconstruct them, these variables contained loss values from 0 % to 50 %.

Palabras Claves

- Aprendizaje máquina (Machine Learning).
- Datos faltantes (Missing Values).
- Imputación de datos.
- Redes neuronales artificiales (ANN).

Capítulo 1

INTRODUCCIÓN

Hoy en día es común utilizar sistemas informáticos para realizar distintas tareas, pero en su mayoría se emplean para capturar información relevante y que a su vez sea posible consultarla y manipularla a según sean los intereses de los usuarios. Por ejemplo los cajeros dentro de las cadenas de los supermercados utilizan un sistema informático que les permite consultar los productos mediante la lectura de su código de barras ya sea introduciéndolo manualmente o escaneado y también les brinda la posibilidad registrar una venta y realizar más tareas que son sumamente útiles para el desempeño de su función. Al final del día toda esta información habrá quedado almacenada en algún servidor y podrá ser consultada por el gerente de la tienda quién muy seguramente tomará decisiones basadas en los resultados de estos datos, gracias a estos se podrán construir gráficas que muestren los productos más vendidos, las horas y los días en que se realizan más ventas, en su caso tambien es posible realizar predicciones de ventas y un sinfín más de cálculos estadísticos que contribuirán a la toma de decisiones. Claro, todo esto es posible gracias a la garantía de que los datos que fueron almacenados de manera correcta y que no existió ningún fallo o anomalía que provocara perdidas de información.

Este es un ejemplo trivial y parece un proceso bastante simple, no deberían de ocurrir fallos a la hora de realizar las capturas de las ventas, al menos que el computador no encienda, deje de funcionar inesperadamente, la red haya sufrido algún desperfecto o en otras palabras que se encuentre caída, se tenga una muy mala latencia o bien, que no exista una conexión establecida para el intercambio de información. Pero esto no ocurre en las grandes cadenas de supermercados, puesto que este es un proceso crucial y por eso mismo existen estrategias para asegurar que los sistemas informáticos siempre estén funcionando de manera correcta y sin fallo alguno.

Más sin embargo existen sistemas informáticos más complejos donde las capturas de datos no son realizadas mediante un usuario, sino que son registradas por las muestras que se obtienen a través de algún dispositivo y en el caso de que alguno de ellos falle la información se habrá perdido y no existirán registros en ese lapso en los cuales ocurrió este suceso. Es totalmente válido y acertado pensar que así como existen estrategias para asegurar el funcionamiento de los dispositivos también se deben considerar en los casos contrarios, por ejemplo, que los datos se envíen de manera periódica en intervalos de corto tiempo para así evitar que se pierda toda la información en caso de que algún dispositivo fallara, o bien cuando ocurra esto utilizar algún otro que se encuentre a disposición para que este tome su lugar y recopile la información y algunas otras más soluciones que imaginemos. Sin embargo, en sistemas como estos se enfrenta a problemas como pueden ser el mal estado del clima por las lluvias lo que puede afectar a las transmisiones de información, así mismo las altas y bajas temperaturas las cuales repercuten en la integridad de los dispositivos y por ende su funcionamiento, de tal manera que en ocasiones resulta un poco difícil asegurar en un 100 % que la captura de información se hará con éxito debido a la naturaleza de los distintos tipos de sistemas informáticos que se utilizan para esto.

Por otro lado, en ocasiones las pérdidas de información tienen que ver con el factor humano al momento del procesamiento de los datos, es decir cuándo se van a capturar los datos. Un ejemplo es cuando se acude al médico, un procedimiento común es que la enfermera atienda primero al paciente y tome algunos datos importantes como lo son peso, estatura y presión arterial, mismos que son parte del historial clínico de este, el cuál puede ser utilizado para predecir si dicho paciente es propenso a enfermedades como lo son la diabetes. Más sin embargo en ocasiones este procedimiento no se realiza provocando así pérdidas de información por cada paciente que asiste a consulta. De esta manera resulta un poco difícil estudiar o analizar el historial clínico de un paciente debido a que no se registró de manera correcta su información, compararla contra la de otros pacientes de la misma edad o peso pudiera dar algún resultado, pero se corre el riesgo de producir resultados sesgados y por ende pocos certeros.

Esto es un problema recurrente en todo sistema informático y siempre se tiene la probabilidad de que existan fallos en el procesamiento de la información y puede ocurrir en los distintos sistemas que se utilizan para los diversos contextos. Como se expuso anteriormente las causas pueden verse influenciadas por el factor humano o por fallos inesperados en los dispositivos utilizados. Generalmente a la información que no está disponible y que podría ser útil para realizar algunos análisis que ayuden y aporten a la toma de decisiones se le denomina valores faltantes (**Missing Values/Data**).

Los valores faltantes entorpecen los procesos de análisis de datos y en problemas del mundo real es sumamente común encontrarse con ellos, Por otro lado, cuando se quiere realizar un modelo de aprendizaje máquina es indispensable no tener la presencia de estos, puesto que las estructuras que permiten realizar este proceso no aceptan valores de este tipo, por lo que deben de ser tratados de manera eficiente para así poner en marcha dichos análisis.

Asi como los datos tienen algún patrón que siguen en su comportamiento, también existen aquellos que representan la relación que hay entre las variables que contienen datos faltantes y se les denomina mecanismos de pérdidas y estos se clasifican en 3: completamente aleatorios (**MCAR**), aleatorios (**MAR**) y no aleatorios (**MNAR**). Los primeros dos están ligados a la probabilidad de que un valor se ausente, en los **MCAR** cualquier dato tiene la misma posibilidad de faltar, de manera que la pérdida no está ligada a ninguna otra variable, por el contrario en los de tipo **MAR**, los datos faltantes depende en si de otra variable, por lo que pueden ser explicados a través de esta dado que su probabilidad de faltar se ve influenciada por esta misma, y los **MNAR** no pueden ser explicados a través de alguna otra variable, ni tampoco tienen la misma posibilidad de ausencia, sino que el motivo de pérdida depende en si de los mismo datos que faltan.

Para este caso de estudio, se presentó el problema de los datos faltantes dentro de un conjunto de datos, el cual ha recopilado desde el año 2012 hasta 2020 muestras en los distintos cuerpos de agua de la República Mexicana, las cuales ayudan a determinar la calidad de esta sustancia, siendo la región de interés el estado de Sinaloa, el cual cuenta con 293 puntos de muestreo.

Los expertos del área se ven afectados puesto que no es posible realizar análisis que contribuyan a este proceso debido a la escasez de la información. Esto por los múltiples problemas que enfrentan tanto como para obtener las muestras y para su procesamiento, siendo esta una de las principales razones por las cuales se tienen pérdidas de información. Dada las circunstancias se llevó a cabo un proceso de reconstrucción de la información sobre estos parámetros, donde se aplicaron técnicas estadísticas y de aprendizaje máquina con el fin de mitigar y dar un tratamiento efectivo a los datos faltantes. Dicho conjunto de datos presentó grandes porcentajes de valores faltantes los cuales van desde un 0 % hasta

un 99 %, por lo cual fue necesario realizar un análisis para la elección de las variables candidatas para el proceso de reconstrucción, de las 256 variables disponibles solo fue posible reconstruir un total de 23 de parámetros reconstruidos, mismos que sus porcentajes de pérdidas fueron inferiores al 50 %.

1.1. Definición del problema

El estudio se enfoca en la reconstrucción de la información de los parámetros que ayudan a determinar la calidad del agua en el estado de Sinaloa. Dicha información ha sido recopilada dentro de los diversos cuerpos de agua de la República Mexicana desde el año 2012 hasta el 2020, esto a través de los distintos monitoreos realizados mediante la Red Nacional de Medición de la Calidad del Agua (**RENAMECA**). Misma que pertenece a la Comisión Nacional del Agua (**CONAGUA**) la cual es el organismo encargado de administrar, regular, controlar y proteger el agua en el país (CONAGUA, 2023). Sin embargo esta información presenta grandes cantidades de datos faltantes, los cuales van desde un rango de 0 % hasta un 99 % y se encuentran dispersos en cada uno de los 256 parámetros dentro de una población total de casi 6,200 registros totales, esto debido a que la obtención de las muestras es un proceso que implica trasladar a un equipo capacitado a cada uno de los 293 puntos de muestreo ubicados alrededor de del estado para así realizar esta actividad. Así mismo, dado que existen diferentes cuerpos de agua, como lo son los lóticos donde el movimiento del agua predomina en una sola dirección, tal es el caso de los ríos, manantiales, arroyos etc. También se tienen aquellos donde el agua permanece en un solo lugar encerrada, a los cuales se les denomina lénticos siendo los lagos y lagunas son un ejemplo de estos. La recolección de muestras en espacios profundos resulta una actividad riesgosa, además para garantizar la integridad de las muestras se necesita de un cuidado especial, como lo es colocarlas en recipientes de ciertos materiales y mantenerlas en el rango de temperatura que la norma indique, así también es necesario llevarlas a laboratorios especializados para que se les hagan sus respectivos procedimientos. De tal manera que esta tarea se vuelve un poco rigurosa y todos estos factores repercuten en el sistema de información encargado de almacenar los datos, lo que da pie a tener grandes cantidades de datos faltantes y siendo estas las principales razones de esta anomalía.

1.2. Hipótesis

Se postula que la aplicación de técnicas basadas en aprendizaje máquina/automático para el manejo de datos faltantes, permitirá obtener resultados más precisos que los métodos tradicionales de imputación y estadística. Esto dado a que las redes neuronales artificiales (ANN) permiten encontrar patrones y tendencias en los datos, apesar de no tener una correlación fuerte entre las variables que se están trabajando y en cambio las técnicas de imputación suelen apoyarse de dicha propiedad y suelen conducir a mejores resultados cuando se tiene una relación evidente. Así, el integrar también el uso de modelos de aprendizaje automático podría resultar en una mejora significativa a la hora de realizar la reconstrucción de los datos faltantes en las muestras de la calidad del agua, lo cual, a su vez, garantizará mejores análisis para la toma de decisiones en este contexto.

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar un sistema capaz de reconstruir información mediante técnicas estadísticas y/o Machine Learning y de esta manera contar con más información para la estimación de parámetros en las muestras en la calidad del agua.

1.3.2. Objetivos Específicos

1. Investigar y seleccionar dentro del estado del arte métodos para la reconstrucción de información con distintas técnicas.
2. Analizar la fuente de información para diseñar estrategias de reconstrucción que se adapten al estado actual de la información.
3. Implementar los métodos y técnicas seleccionadas para comparar su desempeño al reconstruir los parámetros faltantes.
4. Evaluar los resultados y métricas obtenidas a través de la reconstrucción realizada para su posterior interpretación.

1.4. Justificación

La ausencia de datos es un problema recurrente al que todo analista de datos se enfrenta. En cualquier sistema informático siempre existe la posibilidad de que se presenten pérdidas de información, las causas son múltiples, fallos en los dispositivos encargados de recabar datos o errores humanos en la captura de estos, de manera tal que no se puede estar exento de este suceso.

Tal es el caso de la recopilación de las muestras en los cuerpos de agua, este proceso se torna un poco complejo y difícil de manera que produce grandes cantidades de datos

faltantes. Es por ello por lo que la importancia de este proyecto radica en la reconstrucción de la información de estos parámetros que ayudan a determinar la calidad del agua, dado que esta información es sumamente importante para realizar los numerosos análisis que pueden contribuir a la toma de decisiones de los expertos y que favorecen a una mejor determinación de la calidad de esta sustancia, la cual es vital en los seres vivos. Por otro lado, el tener datos faltantes entorpece la generación de modelos de aprendizaje máquina dado que las estructuras que permiten realizar este procedimiento no aceptan valores de este tipo, aunado a esto uno de los principales objetivos de esta es que se aprenda de los mismos datos (Qiong and Ying, 2012) por lo cual el no contar con información completa no resulta favorable cuando se quiere generar dichos modelos. La aplicación de estos es útil para realizar simulaciones en el tiempo y así mismo para estimar posibles valores, trayendo consigo mismo grandes beneficios para el estudio de estos datos, pero se requiere de una fuente de información completa y certera. De manera que la reconstrucción de estos parámetros proveerá nuevas fuentes de información las cuales brindan un abanico de posibilidades para seguir realizando estudios sobre estas muestras. Este proyecto se enfoca solamente en generar nuevas fuentes de información y de probar ciertas estrategias y de esta manera determinar cuáles son más apropiadas para realizar este proceso.

1.5. Estructura de la tesis

El presente trabajo está dividido en seis capítulos y está presentado de la siguiente manera:

- **Capítulo 1 - Introducción:** Se explica la definición del problema, la hipótesis, el objetivo, los objetivos específicos y la justificación.
- **Capítulo 2 - Marco Teórico:** Se describen los conceptos fundamentales para la elaboración del proyecto como lo son, Inteligencia artificial (Artificial Intelligence/AI), Aprendizaje máquina (Machine Learning/ML), Redes neuronales (Neural Networks), Mecanismos de perdidas de datos, Imputación simple y múltiple.
- **Capítulo 3 - Estado del Arte:** Se presenta un resumen de la documentación y las prácticas más relevantes que se han desarrollado y empleado hasta la fecha en el contexto de la investigación abordada en este trabajo.
- **Capítulo 4 - Metodología:** Se plasma el proceso seguido de principio a fin, lo cual constituye una exposición completa de las acciones emprendidas durante el mismo.
- **Capítulo 5 - Resultados:** Se muestran los resultados obtenidos atraves de los procesos de imputacion de datos faltantes y de generacion de modelos de aprendizame maquina para la prediccion de los mismos.
- **Capítulo 6 - Conclusiones:** Se expone un resumen de los hallazgos finales basados en los resultados obtenidos durante la metodología empleada y se mencionan algunas recomendaciones para manejar los datos faltantes en conjuntos de datos.

Capítulo 2

MARCO TEÓRICO

2.1. Inteligencia artificial (Artificial Intelligence/AI)

La concepción de la inteligencia artificial se remonta a los años 90s poco después de la segunda guerra mundial durante el año 1956 (Russell and Norvig, 2009). Uno de los nombres más mencionados e importantes de aquel entonces fue el de Alan Turing (1950), quién fue un matemático de descendencia británica, considerado uno de los fundadores de la informática moderna y de la misma inteligencia artificial.

Turing definió a la IA como “La capacidad que una computadora tiene para lograr un rendimiento a nivel humano en tareas cognitivas” (Ramesh et al., 2004).

Asi mismo, Turing propuso una prueba hoy en día conocida como el “Test de Turing”. La cuál es aprobada, si un humano realiza preguntas escritas a una computadora y este no es capaz de distinguir si las respuestas provienen de una máquina o de una persona. De ser así, entonces se habrá cumplido con éxito la prueba (Russell and Norvig, 2009). Tal como se aprecia en la **figura 2.1**.

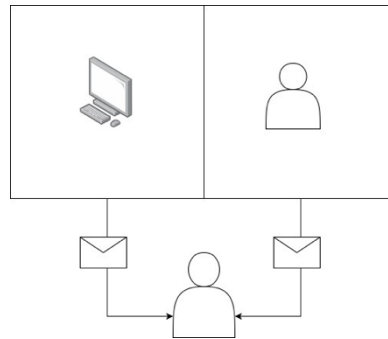


Figura 2.1: Test de Turing

Mientras que (Borana, 2016) define a la inteligencia artificial “Como aquella inteligencia exhibida por una entidad artificial para resolver problemas complejos y que generalmente se supone que tal sistema es una computadora o máquina” Existen dos variantes en la inteligencia artificial: débil y fuerte. Tal cual sus nombres indican la principal diferencia radica en la capacidad que tienen cada una de ellas para realizar distintas tareas.

- **Inteligencia artificial débil (Weak AI)** También conocida como inteligencia artificial estrecha (Narrow AI) es aquella que se encuentra limitada por una área específica o estrecha. Por ejemplo, los algoritmos que pueden completar un conjunto de funciones finitas, como lo son los juegos de ajedrez y de damas chinas (Gee Wah and Wang Chi, 2020).

- **Inteligencia artificial fuerte (Strong AI)**

Inteligencia artificial general (Artificial General Intelligence) es otra manera de nombrar a la inteligencia artificial fuerte, esta es capaz de funcionar como un humano y de poseer inteligencia al mismo nivel. También tiene la capacidad de razonar, comprender y de realizar aquellas tareas complejas que una inteligencia débil no puede resolver (Gee Wah and Wang Chi, 2020).

Algunos ejemplos de esta son, los asistentes de voz, chatbots, automóviles autónomos, robots y entre otros más.

Para que una inteligencia artificial sea capaz de realizar todas esas tareas complejas es necesario que tenga un proceso de aprendizaje, generalmente a esto se le conoce como aprendizaje máquina o automático.

2.1.1. Aprendizaje Máquina (Machine Learning/ML)

“El aprendizaje máquina es aquel estudio científico de los algoritmos y modelos estadísticos que utilizan los sistemas informáticos para realizar alguna tarea, sin la necesidad de ser programada explícitamente” (Mahesh, 2019).

Lo que Mahesh dice, es que no se le necesita decir con instrucciones en un lenguaje de alto nivel (código) al ordenador como realizar dicha programación. Además menciona que “el propósito del aprendizaje máquina es que se aprenda en si de los mismos datos”.

De tal manera, que dicho modelo será capaz de encontrar patrones y tendencias durante el procesamiento de estos mismos, a este proceso se le denomina aprendizaje y se clasifica en dos tipos:

- Supervisado.
- No supervisado.

2.1.2. Aprendizaje Supervisado (Supervised Learning)

(Qiong and Ying, 2012) sostienen que “El aprendizaje supervisado, es un paradigma del aprendizaje máquina para adquirir la relación entre la información de entrada y salida de un sistema basado en un conjunto de muestras de entrenamiento.”

También dice que el objetivo del aprendizaje supervisado es “Construir un sistema artificial que sea capaz de aprender sobre el mapeo entre las entradas y salidas del sistema, y así de esta manera predecir las salidas correspondientes para las nuevas entradas” (Qiong and Ying, 2012).

En la **figura 2.2**, se observa el proceso para realizar la generación de un modelo de aprendizaje máquina, esto de manera supervisada. Para este caso el conjunto de datos o fuente de información debe de estar totalmente categorizado o etiquetado con sus respectivas clases. Después, es necesario separar los datos en subconjuntos; uno para el entrenamiento, el cual generalmente utiliza de un 70 a un 80 de la información total y otro para las pruebas de evaluación que corresponde al resto según sea el caso (Gholamy et al., 2018).

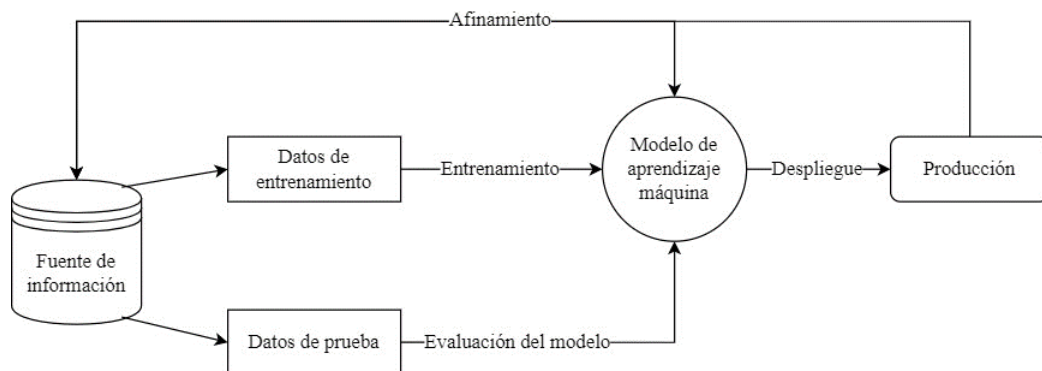


Figura 2.2: Flujo de trabajo de un modelo de ML con aprendizaje supervisado

Una vez separados los conjuntos de entrenamiento y prueba se procede a generar un modelo de aprendizaje máquina provisional, dicho modelo utiliza los datos de entrenamiento y los procesa a según sea el caso de la técnica empleada.

Posteriormente, es necesario realizar una evaluación utilizando los datos de pruebas y los que el modelo ha producido con la finalidad de corroborar la eficiencia de este mismo, esta misma es determina por alguna de las métricas que se utilicen para el modelo que se está desarrollando. De existir un resultado favorable dicho modelo queda listo para ser integrado, caso contrario se realiza un afinamiento. Esto puede ir desde agregar o quitar datos o hasta utilizar uno conjunto de datos totalmente diferente, por eso es muy importante que se realice un preprocesamiento y analizar las técnicas disponibles y así seleccionar la más adecuada.

2.1.3. Aprendizaje no Supervisado (Unsupervised Learning)

A diferencia del aprendizaje supervisado, aquí no se requiere etiquetar o categorizar los datos. El aprendizaje no supervisado tiene la capacidad de agrupar por características similares (**clustering**) a cada uno de los correspondientes datos (Sandoval, 2018).

Por su parte (Dridi, 2021) enuncia que “El aprendizaje supervisado se refiere a los algoritmos que se utilizan para identificar patrones en conjuntos de datos que contienen muestras que no están clasificadas ni etiquetadas”

De manera tal que de un conjunto de datos donde no se conocen las características de cada una de las muestras, gracias a los algoritmos de aprendizaje no supervisado es posible crear grupos con las mismas características, tal y como se muestra en la **figura 2.3.**

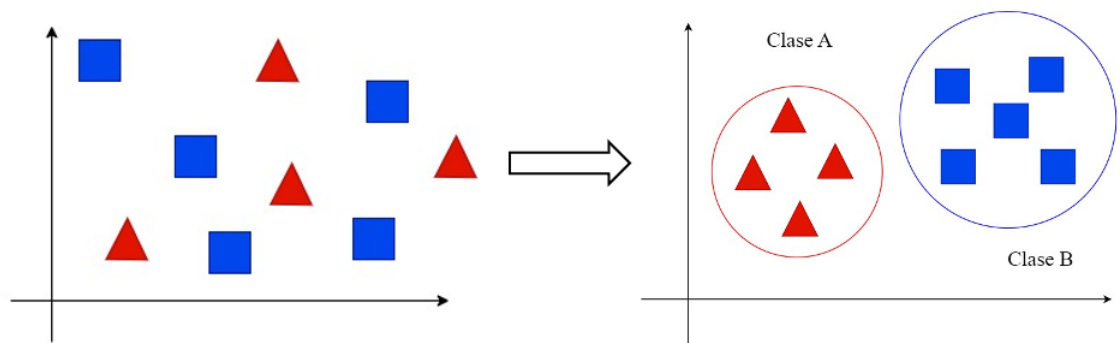


Figura 2.3: Aprendizaje No Supervisado

Los dos tipos de aprendizajes mencionados supervisado y no supervisado, resultan útiles a la hora de resolver problemas donde se busca extraer información de algún conjunto de datos con el fin de realizar tareas de clasificación, reconocimiento o agrupamiento, tanto en fuentes de información categorizadas. Otro punto clave en la generación de modelos de aprendizaje máquina son las redes neuronales. Estructuras que permiten realizar el entrenamiento de dichos modelos a través de los procesamientos internos que realizan gracias a los recursos del ordenador.

Después de seleccionar algún tipo de aprendizaje, también es importante escoger una red neuronal que se adecue al problema que se quiere resolver. Ya que existen distintas vertientes en dichas estructuras y cada una de ellas se puede adaptar mejor a según el caso. A continuación, se mencionan únicamente aquellas que fueron utilizadas en este trabajo.

2.2. Redes Neuronales (Neural Networks)

2.2.1. Red Neuronal Artificial (Artificial Neural Network/ANN)

Las redes neuronales son modelos digitalizados de un cerebro humano, diseñados para simular la manera en que este procesa la información (Agatonovic-Kustrin and Beresford, 2000). A través de los patrones y relaciones entre los datos es como una ANN consigue reunir conocimiento.

De la misma manera (Wu and Feng, 2018) coinciden en que una red neuronal artificial es un modelo computacional que busca simular el procesamiento de las redes neuronales de nuestro cerebro y la forma en la que se lleva a cabo el procesamiento de la información. Dicho modelo se compone de una gran cantidad de nodos o neuronas como lo es en el caso del cerebro humano, conectados entre sí mismos y cada uno de estos representa una función específica de salida, a la cual se le denomina función de activación. Así mismo existe una conexión entre el par de nodos representada mediante un peso (**weight**) la cual es una señal que pasa a través de esta y es equivalente a la memoria de la ANN, las salidas dependen en si de estas conexiones, es decir, de la manera en que se conectan los nodos con sus respectivos pesos.

Las ANN utiliza unidades de procesamiento de neuronas también denominada como capas, y pueden representar distintos objetos, como lo son las características de un conjunto de datos (**features**), imágenes de letras o cualquier otro tipo de patrón de abstracción.

Dichas unidades se dividen en tres capas:

- Capa de entrada (**input**)
- Capa de salida (**output**)
- Capa oculta (**hidden**)

Y en palabras simples se puede describir que la capa de entrada corresponde a los datos extraídos del mundo real, mientras que la capa de salida es el resultado que se obtiene de la red y la capa oculta es una unidad que se ubica entre estas dos y contiene datos de forma diferente que los de entrada, lo cual permite que la red procese más características y no solo las originales.

En la **figura 2.4** se puede apreciar la estructura de una ANN y en esta misma se observan las 3 capas mencionadas anteriormente y la conexión entre ellas.

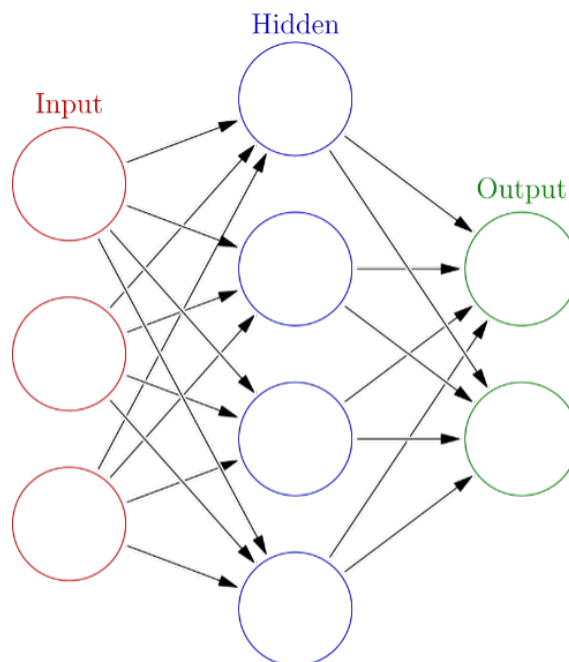


Figura 2.4: Estructura de una ANN

Por su parte (Zupan, 1994) menciona que la descripción más cercana de una ANN es compararla con una caja negra que tiene múltiples variables de entrada (**input**) y de salida (**output**). Las cuales operan utilizando una gran cantidad de datos conectados entre sí mismos, tal como se muestra en la figura 2.5.

Así mismo, enuncia que no es necesario que dichas variables tengan algún tipo de correlación lineal y que usualmente este tipo de redes neuronales conducen a buenos resultados para estos escenarios.

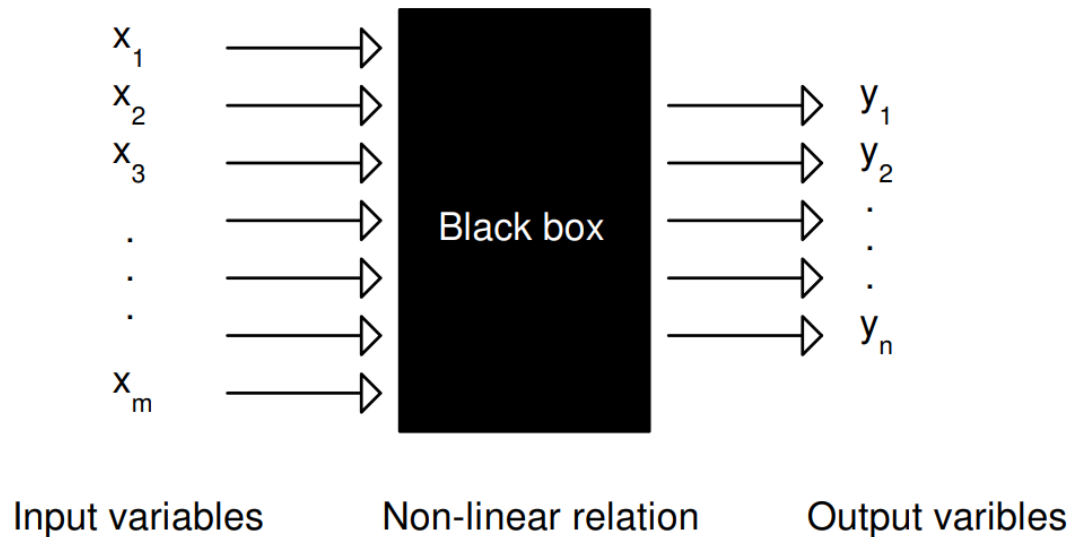


Figura 2.5: ANN: Caja negra
(Zupan, 1994)

2.2.2. Red Neuronal Convolutacional (Convolutional Neural Network/CNN)

Por otro lado, una red neuronal convolutacional (**CNN**) es un modelo de aprendizaje profundo (**Deep Learning**) para el procesamiento de datos que tienen un patrón de tipo cuadrado (Yamashita et al., 2018), como lo es en el caso de las imágenes. Las cuales se pueden descomponer en un matriz con los valores en blanco y negro, tal y como se muestra en la **figura 2.6**, donde cada uno de ellos representa los pixeles que conforman dicha imagen. Sus principales usos están destinadas a la visión por computadora para la clasificación e identificación de características particulares de una imagen. Si bien estas redes se emplean para dichas funciones, se toman a consideración puesto que son capaces de procesar patrones en forma de matrices y así mismo para evaluar su desempeño en los modelos de reconstrucción, y así determinar si son útiles o no.

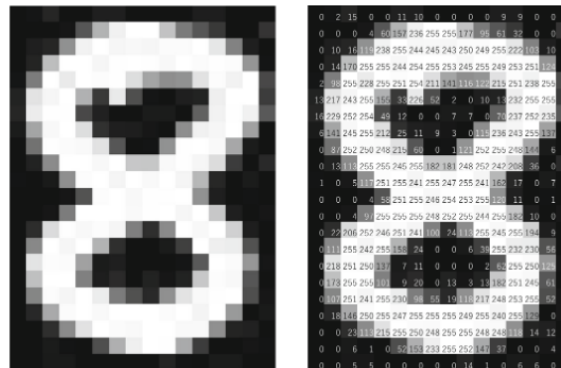


Figura 2.6: Patrón de tipo cuadrado
(Yamashita et al., 2018)

Al igual que las **ANN**, las redes neuronales convolucionales utilizan distintas capas, pero en este caso las **CNN** además de las capas de entrada y salida también posee las siguientes:

- **Capa de convolución (Convolutional Layer)**
- **Capa de agrupación (Pooling Layer)**
- **Capas completamente conectadas (Fully Connected Layers)**

La principal diferencia entre una **ANN** y una **CNN** es que en la primera cada neurona está conectada entre sí, mientras que en una red neuronal convolucional solo la última capa se encuentra completamente conectada (Gogul and Sathiesh Kumar, 2017).

2.3. Métricas de evaluación

2.3.1. R-Cuadrado (Coeficiente de Determinación)

Correlación Lineal

La correlación lineal se refiere al grado de variación que existe entre dos o más variables (Senthilnathan, 2019). Por ejemplo, el precio de una casa seguramente se verá influenciado por las dimensiones del terreno, número de habitaciones, baños, etc... De tal manera que existe una correlación lineal entre dichas variables.

Existen dos principales tipos de correlación positiva y negativa o inversa. Una correlación lineal positiva indica que los valores de ambas variables varían de manera similar. Mientras que una correlación lineal negativa, denota que dichos valores varían de forma inversa. La manera más sencilla para identificar si dos variables tienen alguno tipo de correlación es utilizando un diagrama de dispersión (de Madrid, 2023). En la **figura 2.7** se ilustran los grados de correlación entre dos variables denotados por R , dicha expresión representa el coeficiente de correlación.

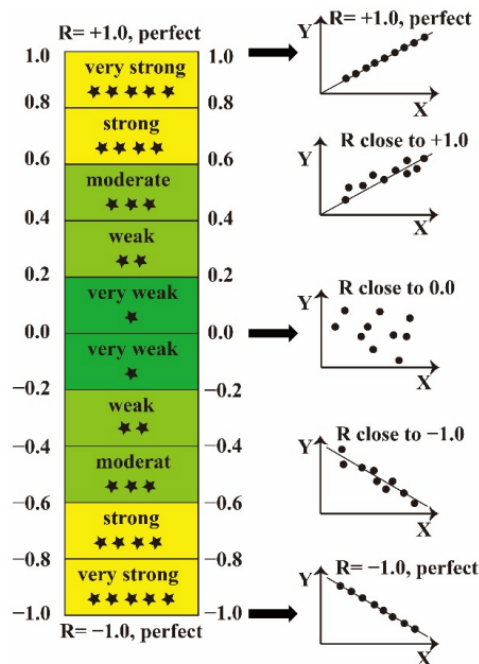


Figura 2.7: Grados de correlación lineal
(Jierula et al., 2021)

Los rangos de R van desde $[-1, 1]$ y estos se interpretan de la siguiente manera:

1. Cuando se tiene un valor de -1 o 1 , indica que existe una correlación perfecta entre ambas variables.
2. Un valor de -0.8 o 0.8 representa una correlación fuerte, ocurre de la misma manera si el valor se encuentra entre -0.9 o 0.9 .
3. Un rango de -0.4 a 0.6 o de 0.4 a 0.6 indica una correlación moderada.
4. Para los valores de R que están desde -0.2 y 0.4 o bien de 0.2 a 0.4 , se dice que la correlación es débil.
5. Una correlación muy débil se da cuando los valores oscilan entre por debajo de -0.2 y 0.2 y un valor de 0 indica una correlación nula.

Por otro lado, el coeficiente de determinación R^2 es una métrica basada en correlación y se utiliza ampliamente para evaluar que tan buenos son los modelos de regresión utilizando ANNs en base a un ajuste lineal (Jierula et al., 2021)

R^2 corresponde al cuadrado del coeficiente R , dicho coeficiente puede ser expresado por la correlación de Pearson, la cual representa el grado de relación entre las variables actuales y las que fueron pronosticadas o predichas.

En la **ecuación 2.1**, se muestra la fórmula para obtener el coeficiente de Pearson.

$$R = \frac{c[x, y]}{\sqrt{S_x S_y}} \quad (2.1)$$

Donde $c[x, y]$ corresponde a la media que existe entre las diferencias de los productos de los valores actuales y los pronosticados.

$$c[x, y] = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i) (y_i - \hat{y}_i)$$

S_x el MSE (véase la **ecuación 2.3**) de los valores actuales.

$$S_x = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2$$

S_y el MSE de los valores pronosticados.

$$S_y = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Mientras que en la **ecuación 2.2**, se aprecia una manera simplificada para calcular R^2 una vez obtenido R .

$$R^2 = 1 - \frac{S_x}{S_y} \quad (2.2)$$

2.3.2. Error Cuadrático Medio (Mean Squared Error/MSE)

Por su parte, el **MSE** se encarga de determinar el error cuadrático medio entre los valores actuales y los obtenidos por el modelo.

Esto lo hace midiendo en forma vertical la distancia que existen entre el valor real y el pronosticado y se calcula el cuadrado de esta diferencia. Este procedimiento se realiza para cada una de las muestras y posteriormente se divide entre el número de estas, es decir se obtiene su media (Jierula et al., 2021).

En la **ecuación 2.3** se muestra la fórmula para obtener el Error Cuadrático Medio (MSE)

$$MSE = \frac{1}{n} \sum_{i=1}^n \left(Y_i - \hat{Y}_i \right)^2 \quad (2.3)$$

Donde:

n representa el número de muestras

Y_i al valor real y

$\hat{Y}_i$ el valor pronosticado.

El rango de los posibles valores del **MSE** oscila entre $(0, \infty)$. Cuando se obtuvo un valor cercano a cero, esto indica una alta exactitud (**accuracy**) en el modelo de predicción.

Por lo tanto, entre más cercano sea el valor del **MSE** a cero se obtendrán mejores resultados para la predicción. Esta métrica es utilizada de manera predeterminada como función de pérdida (**loss function**) en la generación de modelos de aprendizaje máquina, tal y como se muestra en los capítulos de desarrollo y resultados.

2.3.3. Desviación Cuadrática Media (Root Mean Squared Error/RMSE)

La desviación cuadrática media o también llamada media cuadrática (RMSE) determina la magnitud promedio del error que existe entre el valor real y el pronosticado (Jierula et al., 2021).

En otras palabras, es la distancia promedio medida de manera vertical entre dichos valores y no es más que la raíz del MSE, tal y como se puede apreciar en la **ecuación 2.4**.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2} \quad (2.4)$$

De la misma manera que el **MSE**, los rangos se encuentran en el intervalo de $(0, \infty)$ y entre más cercano a 0 se encuentre el valor del **RMSE** mayor será la precisión del modelo de predicción.

Existen algunas otras métricas como lo es el error absoluto medio (**MAE**), que al igual que el **RMSE** expresa la magnitud promedio del error de predicción con los valores reales, pero lo hace de manera absoluta, tal y como se muestra en la **ecuación 2.5**.

$$MAE = \frac{1}{n} \sum_{i=1}^n |Y_i - \hat{Y}_i| \quad (2.5)$$

2.4. Datos Faltantes (Missing Data)

2.4.1. Mecanismos de perdidas

Rubín (1976) y Little (1987) definieron y asentaron las bases de la teoría sobre los mecanismos de pérdidas en los datos y los clasificaron en tres categorías:

- Datos Faltantes Completamente Aleatorios (**Missing Data Completely At Random/MCAR**)
- Datos Faltantes Aleatorios (**Missing Data At Random/MRA**)
- Datos Faltantes No Aleatorios (**Missing Data Not Random/MNRA**)

Los mecanismos de perdidas representan la relación estadística entre las variables y la probabilidad de los datos faltantes (Nakagawa and Sosa, 2015).

(Salgado et al., 2016) enuncian que es posible describirlos a través de la relación entre los datos observados o disponibles y de aquellos que no son observables, es decir los faltantes.

Para definir los mecanismos de perdidas en términos matemáticos, dichos autores señalan que se le denomine X al conjunto de datos completo y que este se componga de otros dos conjuntos. Uno que represente a los datos que están disponibles (observados) y otro para los faltantes (**missing**).

De manera tal que dicho conjunto puede ser representado con la siguiente expresión:

$$X = \{X_o, X_m\} \tag{2.6}$$

Donde X_O representa a los datos observados o disponibles y X_m aquellos que no están disponibles o bien son faltantes.

Adicionalmente, se define la siguiente regla. La cual asigna el valor de 1 a $\mathbf{R}$ si el dato está disponible y en el caso contrario, $\mathbf{R}$ asumirá el valor de 0.

$$R = \begin{cases} (1, & \text{si } X \text{ disponible}) \\ (0, & \text{si } X \text{ faltante}) \end{cases} \quad (2.7)$$

De tal manera, que los mecanismos de perdidas pueden ser representados y entendidos en términos de la probabilidad que un valor falte, es decir:

$$\Pr(R \mid x_o, x_m) \quad (2.8)$$

2.4.2. Datos Faltantes Completamente Aleatorios (Missing Data Completely At Random/MCAR)

La ausencia de valores no se relaciona con ninguna otra variable. Es decir, que los datos faltantes que son completamente aleatorios (**MCAR**) tienen la misma probabilidad de faltar, por lo tanto, se puede asumir la siguiente equivalencia:

$$\Pr(R \mid x_o, x_m) = \Pr(R) \quad (2.9)$$

Por ejemplo, si algún médico se olvidará de capturar la edad de los primeros 10 pacientes, no existe una variable que sea capaz de describir la ausencia de estos valores y no depende de ninguna de las características del paciente. Simplemente que el doctor olvidó registrar estos datos.

2.4.3. Datos Faltantes Aleatorios (Missing Data At Random/MRA)

En el caso de los datos faltantes aleatorios (**MRA**) estos no tienen la misma probabilidad de faltar. Sino que esta depende de los datos observados, de tal manera que la probabilidad queda dada por la siguiente expresión:

$$\Pr(R \mid x_o, x_m) = \Pr(R \mid x_o) \quad (2.10)$$

Retomando el caso del médico, si por algún motivo este también se olvidará de registrar el peso del paciente, la ausencia de información acerca de su IMC (índice de masa corporal) podría ser explicada a través de dicha variable, ya que esta es necesaria para realizar dicho calculo.

2.4.4. Datos Faltantes No Aleatorios (Missing Data Not Random/MNRA)

Estos datos no tienen la misma posibilidad de faltar (**MCAR**), ni tampoco se puede explicar la ausencia de su información mediante otra variable tal como es el caso de los **MRA**, sino que esta depende del valor de los datos no observados.

(Goldberg et al., 2021) mencionan que generalmente los datos **MCAR** y **MRA**, son relativamente más sencillos de manejar en comparación de los datos **MNRA**.

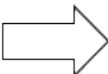
2.5. Técnicas y métodos para el tratamiento de datos faltantes

(Hartini, 2017) y (Kang, 2013), comparten y sugieren las siguientes técnicas para el manejo y tratamiento de datos faltantes:

2.5.1. Eliminación Por Lista (List-wise Deletion)

Es una de las técnicas más utilizadas para el tratamiento de datos faltantes, también se le denomina **caso de análisis casos completo** (Nakagawa and Sosa, 2015).

El procedimiento consiste en eliminar aquellos casos o filas donde se tengan valores faltantes, tal y como se muestra en la **figura 2.8**.



genero	edad	resultado
Hombre	20	Positivo
Mujer		Negativo
Mujer	30	Positivo
	28	Negativo
Mujer		Positivo
Hombre	25	Positivo
Hombre	21	Positivo

id	genero	edad	resultado
1	Hombre	20	Positivo
3	Mujer	30	Positivo
6	Hombre	25	Positivo
7	Hombre	21	Positivo

Figura 2.8: Eliminación por lista

Lo más ideal es aplicarla en aquellos datos que tengan exactamente la misma posibilidad de ausencia, es decir en los **MCAR** (Heymans and Twisk, 2022).

En el capítulo de metodología se detalla cómo se implementó esta técnica, la cual resultó útil para obtener un conjunto de datos sin valores faltantes. Mismo que se utilizó para generar modelos de imputación y de la misma manera dicho conjunto proporcionó la posibilidad de evaluar y determinar que precisos fueron los resultados obtenidos y así mismo para los modelos de aprendizaje máquina.

2.5.2. Imputación Simple

La imputación simple consiste en generar posibles valores para aquellos casos donde se tiene ausencia de información y así posteriormente puede realizar análisis como si no faltara ningún dato (Allison, 2009). Estas técnicas consisten en reemplazar los datos faltantes por un nuevo valor obtenido de una estimación de la distribución de la misma variable donde se presentan estos casos y lo hacen utilizando una sola estimación (Donders et al., 2006) a través de alguna estrategia como la imputación por media. La cual consiste en reemplazar los valores faltantes con la media obtenida de los valores conocidos, es una técnica sencilla pero no es recomendada aplicarla en conjuntos de datos donde se presenten grandes cantidades de pérdidas ya que al reemplazar siempre con un único valor se estaría cambiando la distribución de los datos originales.

2.5.3. Regresión Lineal

(Bar, 2017) Menciona que para aplicar regresión lineal con la finalidad de manejar o darles un tratamiento a los datos faltantes se debe generar un modelo para cada variable que cuenta con dicha característica y que esta misma se utilice como variable dependiente. Y que aquellas donde se tenga un grado de correlación con la misma sean empleadas para dicha predicción. También deja saber que es indispensable que las variables independientes no tengan valores faltantes y que esta técnica funciona mejor que las estrategias de imputación por media o moda.

2.5.4. Imputación Múltiple

A diferencia de la imputación simple, aquí se realizan distintas estimaciones para determinar el valor mediante distintos algoritmos como el MICE (**Multiple Imputation by Chained Equations**) el cual se refiere a la imputación múltiple por ecuaciones encadenadas. Dicho algoritmo cada variable tiene un propio modelo de imputación y en general es más potente que la imputación simple.

(Little et al., 2014) Mencionan que se suele confundir el termino de imputar datos, con inventar datos, particularmente hacen énfasis en que un imputación simple o sustitución por media si lo es. Esto debido que, siempre se reemplaza el mismo con el mismo valor y no se puede tener la certeza de que dicho valor es exactamente igual al dato no observable.

2.5.5. Algoritmo K Vecinos Más Cercanos (K-Nearest Neighbor/KNN)

El algoritmo KNN permite realizar imputaciones múltiples de manera iterativa, está diseñado para encontrar k vecinos más cercanos para un dato faltante utilizando los datos cercanos a este, los cuales si están completos y lo reemplaza con el valor más frecuente (Zhang, 2012).

A través de e la matriz de la distancia Euclidiana es como este algoritmo lo determinar dicho vecino más cercano, al ignorar los valores que faltan y aumentar el peso de las coordenadas que no faltan, se determina la distancia euclidiana (Juna et al., 2022). En la **figura 2.9** se puede ver un ejemplo del algoritmo KNN.

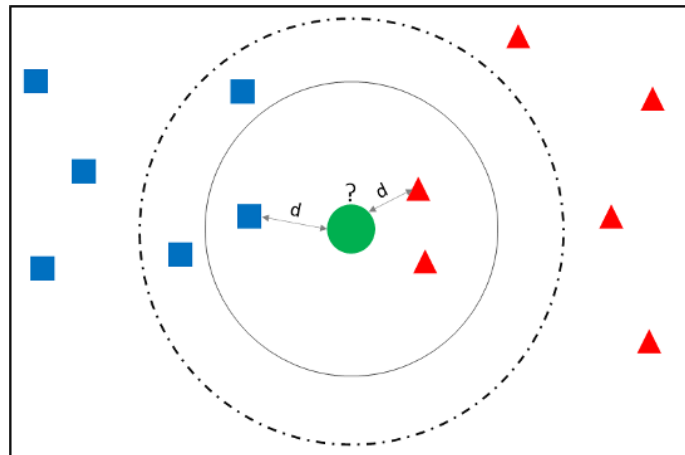


Figura 2.9: Algoritmo KNN

La fórmula para determinar se muestra en la **ecuación 2.11**

$$D_{xy} = \sqrt{\text{peso} \cdot (\text{distancia de las coordenadas})^2} \quad (2.11)$$

$$\text{peso} = \frac{\text{número de coordenadas}}{\text{número de coordenadas presentes}}$$

2.6. Tecnologías y recursos

2.6.1. Python

Python es lenguaje de programación de alto nivel, propósito general y multiparadigma. Fue desarrollado por el neerlandés **Guido Van Rossum** alrededor de los años 90's. Es comúnmente utilizado para el desarrollo de software, en aplicaciones web, para la ciencia de datos (**DS**) y dentro del aprendizaje máquina (**ML**).

En (Amazon, 2023) se menciona que los desarrolladores utilizan este lenguaje dado que es eficiente y fácil de aprender, además también porque se puede ejecutar en distintas plataformas, es un software gratis, se integra bien a los diferentes sistemas y aumenta la velocidad en el desarrollo. Así mismo, Python proporciona una gran variedad de paquetes útiles para el desarrollo de modelos de ciencia de datos y aprendizaje máquina. Todo esto ha ayudado a posicionar a este lenguaje como uno de los más utilizados para estos temas.

Como todo lenguaje de programación, Python ha evolucionado integrando nuevas funcionalidades en cada versión, en la actualidad su versión más reciente es la 3.11.4 (Python, 2023). En la **figura 2.10** se muestra una línea del tiempo donde se pueden apreciar las distintas versiones de Python.

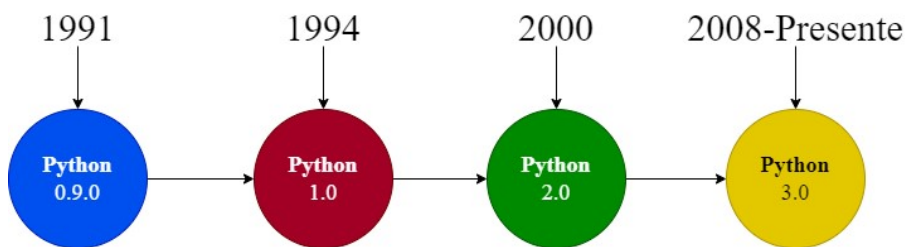


Figura 2.10: Versiones Python

Python 0.9.0: Fue la primera versión y en esta se incluían algunos tipos de datos y funciones útiles para el manejo de errores.

Python 1.0: Trajo consigo nuevas funciones para procesar fácilmente las listas de datos, como lo son asignación, filtros y reducción.

Python 2.0: Sus mejoras fueron enfocadas en la compatibilidad con los caracteres Unicode, así mismo se agregó una forma más sencilla para iterar y recorrer una lista.

Python 3.0: Esta versión incluía nuevas características como la función de impresión, más soporte para la división de números y otras útiles para la gestión de errores.

Características de Python

Algunas características del lenguaje Python son las siguientes:

- **Es un lenguaje interpretado**

A diferencia de un lenguaje compilado, en Python se ejecuta línea por línea el código y de existir algún error la ejecución del programa será detenido.

- **Sencillo y fácil de utilizar**

En Python no se utilizan llaves, aquí los bloques son separados por 4 espacios de indentación (sangría).

- **Es un lenguaje tipado dinámicamente**

No es necesario definir el tipo de variable que se va a utilizar. Python es capaz de determinar en tiempo de ejecución a que tipo de dato corresponde dicha variable.

- **Es un lenguaje multiparadigma**

Está enfocado al paradigma de la programación orientada a objetos (POO) ya que, en Python, todo se considera un objeto. También soporta otros paradigmas como lo son la programación estructurada y funcional.

Bibliotecas y frameworks

Una biblioteca o librería es un conjunto de programas que contienen códigos de funciones que generalmente son utilizados de manera frecuente cuando se está desarrollando software. Estas mismas pueden ser importadas para ser utilizadas por los desarrolladores sin la necesidad de volver a escribir ese fragmento de código, siendo esta una de las principales ventajas por la cual se recomienda crear y utilizar bibliotecas. (Amazon, 2023)

En Python ya se incluye una biblioteca base, donde se encuentran cientos de funciones que son reutilizables. Así mismo, hay más de 137,000 bibliotecas disponibles que se pueden utilizar tanto en el desarrollo web, ciencias de datos o aprendizaje máquina.

Algunas de las bibliotecas más comunes y populares son las siguientes:

- **Matplotlib:** Es útil para la visualización de datos, permite realizar gráficos en 2 y 3 dimensiones.
- **Pandas:** Permite leer estructuras complejas de datos en forma de tablas, brinda la posibilidad de filtrar, combinar, agrupar datos, leer y escribirlos.
- **NumPy:** Se enfoca a las operaciones del álgebra lineal con matrices y vectores.

Más adelante son descritas estas bibliotecas y algunas otras más que fueron utilizadas a lo largo del desarrollo de este proyecto.

Por otro lado, los marcos de trabajo (**frameworks**) son colecciones de paquetes y módulos. Un módulo es un conjunto de clases o códigos que están directamente relacionados, mientras que un paquete es un conjunto de módulos (Amazon, 2023).

En general los marcos de trabajo se utilizan para desarrollar aplicaciones de una manera más rápida, ya que proporcionan una estructura base estandarizada para que los desarrolladores puedan complementarlas y no preocuparse por construir desde cero algunas funciones.

Existen dos tipos de macros en Python:

- **Marco de pila completa:** Proporciona una gran variedad de funcionalidades que permiten crear una aplicación robusta.
- **Micromarco:** Solo proporciona funcionalidades mínimas y básicas para el desarrollo de una aplicación simple y así mismo es posible integrar extensiones en caso de que dichas aplicaciones requieran de funciones más sofisticadas.

Por otro lado, los frameworks más conocido de Python son:

- **Django:** Es un marco de trabajo de pila completa para el desarrollo de aplicaciones web, algunas de las características que este marco incluye son servidores web para el desarrollo y pruebas, también es posible utilizar plantillas del lado del **front-end**.
- **Flask:** Permite desarrollar aplicaciones y servicios web como lo son las API REST que generalmente van del lado del **back-end**, las cuales son muy útiles para el intercambio de datos con un servidor.

2.6.2. scikit-learn

Es un paquete (biblioteca) de Python, que proporciona los algoritmos de aprendizaje máquina más recientes y utilizados dentro del estado del arte (**state-of-the-art**) para la solución de problemas donde se emplea aprendizaje máquina supervisado y no supervisado (Pedregosa et al., 2011).

Fue desarrollada por David Cournapeau en el año 2007 como un proyecto **Google Summer of Code** y posteriormente en el año 2008, Matthieu Brucher continuó con el desarrollo del proyecto incluyéndolo en su trabajo de tesis. Para el año 2010 el desarrollo de Sckit-learn pasó a manos de INRI, donde Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort y Vincent Michel asumieron el liderazgo de dicho proyecto y para el 1ro de febrero del mismo año ya habían lanzado al público la primera liberación (**release**) (Sckit-learn, 2023). Hoy en día, la versión más reciente es la 1.4 (Sckit-learn, 2023) y generalmente se tienen liberaciones de nuevas versiones en un lapso de 3 meses y gracias a que es de código abierto la comunidad contribuye a estas mejoras.

scikit-learn se compone de otros tres paquetes; **NumPy**, **SciPy**, **Matplotlib**. En la **figura 2.11** se aprecia la estructura de las dependencias de la biblioteca.

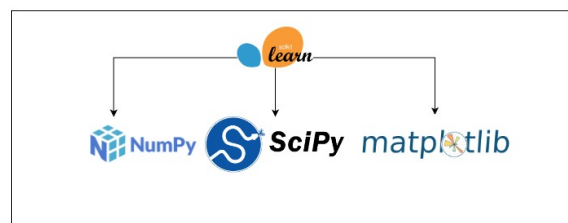


Figura 2.11: Dependencias de **scikit-learn**

Algunas de los módulos y funciones que fueron utilizadas para el desarrollo de este proyecto se listan a continuación:

- **sklearn.metrics:** Proporciona las funciones necesarias para calcular **R²** y **MSE**.
- **sklearn.model_selection:** Ofrece la posibilidad de dividir los conjuntos de datos en entrenamiento y prueba.
- **sklearn.impute:** Provee el algoritmo **KNN** y así mismo otros de imputación simple.

Las especificaciones de cada una de las funciones mencionadas, puede ser consultada en la documentación de **scikit-learn** ubicada en el sitio oficial de la librería, donde también se puede encontrar una gran variedad de ejemplos haciendo uso de estas mismas.

Sitio oficial de la librería: <https://scikit-learn.org/>

2.6.3. Matplotlib

Matplotlib es un paquete (biblioteca) de gráficos 2D utilizado en Python para el desarrollo de aplicaciones mediante secuencias de comandos interactivas que permiten generar gráficos de calidad, los cuales pueden ser publicados dentro de las interfaces de usuarios y en los sistemas operativos (Hunter, 2007).

Fue desarrollada y creada por John D. Hunter (1968–2012) entre el año 2021, cuando este se encontraba desarrollando una aplicación para analizar señales de electrocorticografía (**ECoG**) registradas mediante datos de imágenes médicas en 3D, puesto que Hunter se desenvolvía en sus días como un neurobiólogo de profesión.

1 Electrocorticografía: “es la práctica de utilizar electrodos colocados directamente sobre la superficie expuesta del cerebro para registrar la actividad eléctrica de la corteza cerebral” (Wikipedia, 2023a).

En la referencia mencionada anteriormente, misma que el autor escribió en el año 2007, menciona que durante dicho proyecto se encontraba en un gran debate sobre si utilizar Python o Matlab para su desarrollo. Esto debido a que generalmente, solía combinar Matlab, Fortran y C++ para su flujo de trabajo numérico y estadístico, y a su vez pasaba el tiempo aprendiendo Python como una manera de posponer sus respectivas tareas en las etapas finales del desarrollo de su tesis.

Hunter se inclinó por Matlab, debido a los excelentes gráficos que se manejan aquí (de los cuales Python carecía) y porque en la comunidad **ECoGes** sumamente utilizado este software. Mas sin embargo, la aplicación fue demandado e incrementando en complejidad y funcionalidad, dicha elección habría salido mal en aquel entonces, así que sin más opciones decidió comenzar de nuevo el desarrollo de la aplicación ahora utilizando Python y de esa manera se concibió una de las librerías más utilizadas y conocidas en dicho lenguaje.

Crear gráficos en **Matplotlib** resulta sencillo, puesto que solo se necesita importar la librería, tener una colección con datos y pedirle a la misma que los grafique (**plotting**). En la **figura 2.12** se muestra el resultado de la ejecución de las siguientes líneas de **código** en Python.

Código:

```
import numpy as np
import matplotlib.pyplot as plt

# Generar valores aleatorios para el eje x en el rango de -2 a 2
x = np.linspace(-2*np.pi, 2*np.pi, 100)

# Calcular el valor del seno para cada punto en el eje x
y = np.sin(x)

# Crear la gráfica
plt.plot(x, y)

# Agregar etiquetas y título
plt.xlabel('x')
plt.ylabel('sin(x)')
plt.title('Gráfica de la función seno')

# Mostrar la gráfica
plt.show()
```

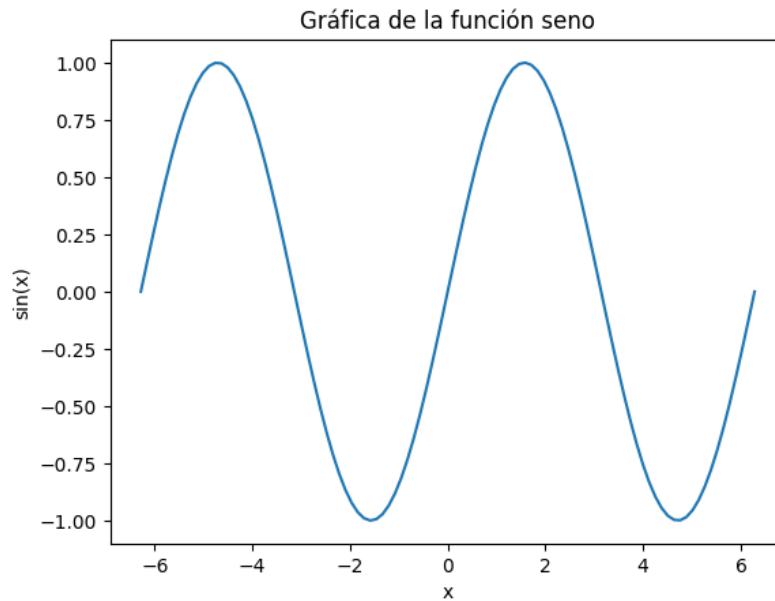
Resultado:

Figura 2.12: Gráfica de la función seno con **Matplotlib**

Gracias a este paquete, es posible desplegar los resultados de los entrenamientos de los modelos de aprendizaje máquina de manera gráfica, lo cual proporciona una visión más general y una mejor comprensión sobre el comportamiento de dichos modelos durante el proceso de entrenamiento. En el siguiente capítulo se detalla acerca de esto y se muestra cómo se empleó dicha librería para ello.

Más ejemplos y especificaciones sobre los módulos de **Matplotlib** pueden ser consultados en el sitio oficial de la librería.

Sitio oficial de la librería: <https://matplotlib.org/>

2.6.4. Pandas

El paquete pandas cuyo nombre se deriva de la palabra en ingles **panel data** (McKinney, 2010). Es útil para la manipulación y análisis de los datos y comúnmente se le denominada el “**Excel**” de Python ya que brinda herramientas estadísticas y algunas estructuras de datos como lo son los DataFrame. Mismos en los que es posible cargar y almacenar en memoria conjuntos de datos de manera tabular y también manipularlos para acceder a su información de manera indexada mediante sus columnas.

El desarrollo de esta librería comenzó en el año 2008 en AQR Capital Management y para finales del próximo año paso a convertirse en software libre (open source), lo que quiere decir que está abierto a la comunidad, mismos que contribuyen a la mejor del desarrollo de esta. Desde el año 2015 **pandas** es financiado por la empresa NumFOCUS misma que ayuda y asegura el éxito del desarrollo de esta librería como proyecto de código abierto a nivel mundial (Pandas, 2023).

A continuación se listan los sucesos más importantes, constituyendo así su línea del tiempo:

- 2008: Comienzo del desarrollo de pandas.
- 2009: pandas pasa a ser **open source**.
- 2012: Se publica la primera edición de *Python for Data Analysis*.
- 2015: pandas se convierte en un proyecto patrocinado por NumFOCUS.
- 2018: Se realizó el primer sprint de desarrolladores en persona.

Algunas características de pandas son:

- Permite manipular datos mediante objetos **DataFrame** de manera rápida y eficiente gracias a que integra indexación.
- Herramientas para la lectura y guardado de datos ya sea en estructura almacenadas en memoria o en distintos formatos como solo: archivos CSV y de texto plano, Microsoft Excel, bases de datos SQL y el formato rápido de HDF5.
- Alineación inteligente de datos y un manejo integrado para los datos faltantes (**Missing Data**).
- Es posible insertar y eliminar columnas de las estructuras de datos, permitiendo así mutabilidad en el tamaño del conjunto de datos.
- Un alto desempeño para la fusión y unión de conjuntos de datos.

Sitio oficial de la librería: <https://pandas.pydata.org/>

2.6.5. NumPy

Su nombre es un acrónimo de **Numerical Python** y es un paquete fundamental para la computación científica en Python (McKinney, 2013). El tipo de estructura base en **NumPy** son los **arrays** los cuales son vectores multidimensionales, también denominados tensores (Harris et al., 2020) donde se pueden almacenar datos numéricos como lo son todos los píxeles de una imagen. Debido a que esta librería utiliza vectores es posible emplear muchas de las transformaciones y operaciones del Álgebra Lineal.

En su mayoría estas estructuras se utilizan para almacenar los datos de entrenamiento y pruebas cuando se realizan los modelos de aprendizaje automático, así mismo es posible manejar grandes cantidades de datos proporcionando un desempeño

bastante favorable, dado que son mas eficientes que las estructuras de datos primitivas de Python.

Sitio oficial de la librería: <https://numpy.org/>

2.6.6. TensorFlow

Es una biblioteca de código abierto (**open-source**) la cual permite implementar algoritmos de aprendizaje máquina. Fue desarrollada originalmente por el equipo de Google Brain, estos mismo a principio del año 2011 construyeron un sistema basado en redes neuronales de aprendizaje profundo llamado DistBelief el cual era meramente de tipo propietario, tuvo aceptación y un rápido crecimiento tanto en compañías como en investigaciones (Wikipedia, 2023b). Posteriormente para el año 2015 nace Tensorflow como una mejora del sistema anterior, su nombre se deriva del uso de **tensores** los cuales son la principal herramienta en la construcción de modelos de aprendizaje máquina (Álvarez, 2022). Esta librería es sumamente utilizada para el aprendizaje profundo (**Deep Learning**).

Sitio oficial de la librería: <https://www.tensorflow.org/>

Capítulo 3

ESTADO DEL ARTE

3.1. Trabajos relacionados

Se llevó a cabo una investigación para explorar el estado actual de la tecnología y las investigaciones relacionadas con la detección y clasificación de deterioros en pavimentos flexibles, con el objetivo de adquirir un conocimiento sólido y actualizado sobre los aspectos teóricos relevantes. Esto implicó revisar documentación y prácticas de proyectos anteriores, artículos científicos, libros, revistas y videos relacionados. También se hizo una evaluación y estudio de los conceptos considerados clave para el desarrollo del proyecto y la redacción del documento. Este aprendizaje proporcionó una base sólida y fundamentada para el desarrollo y la redacción del proyecto, asegurando que se tuvieran en cuenta los conceptos adecuados y se estableciera un fundamento teórico sólido para el trabajo realizado.

Entre las investigaciones revisadas, se destacan aquellos que han sido recopilados y detallados en la **Tabla 3.1**. Esta presenta de manera concisa el título de cada trabajo, presentando en pocas palabras la estrategia y/o herramientas usadas y las conclusiones resultantes de cada estudio. Cada entrada en la tabla plasma el título del trabajo para entender la naturaleza del estudio. Posteriormente, se expone de manera breve

lo acontecido y/o usado en la elaboración. Además, las conclusiones obtenidas para cada trabajo se resumen de manera que se destacan los resultados más clave y las implicaciones derivadas de la investigación. Este enfoque se ha adoptado con el propósito de proporcionar una visión breve y accesible de la esencia de cada investigación, centrándose en los elementos más significativos, ofreciendo una comprensión clara de la amplitud de la misma.

Nombre del trabajo	Estrategias utilizadas	Resultados
Classification of missing values handling method during data mining: Review. (2017)	Listwise Deletion, Pairwise Deletion, Imputación Simple: Media/Mediana/Moda/Regresión, Imputación Múltiple, KNN, Árboles de Decisión y Redes Neuronales	Los métodos de imputación simple son fáciles de implementar, pero no precisos. Se obtienen mejores resultados con técnicas de machine learning.
Data missing not at random in mobile health research: Assessment of the problem and a case for sensitivity analyses. (2021)	Imputación por Regresión, Imputación Múltiple	La Imputación por Regresión es útil para manejar datos MNRA (no aleatorios). La Imputación Múltiple permite obtener predicciones más robustas.
Handling missing data in clinical research. (2022)	Listwise Deletion, Imputación Simple: Media/Mediana/Moda, Imputación Múltiple	La Imputación Múltiple es más efectiva que los métodos de imputación simple, mejorando la validez de los resultados.
Water quality prediction using KNN imputer and multilayer perceptron. (2022)	KNN Imputer, Perceptrón Multicapa (MLP)	Con KNN Imputer se mejoró la calidad del conjunto de datos, permitiendo al Perceptrón Multicapa realizar predicciones más precisas sobre la calidad del agua.

Tabla 3.1: Estado del Arte sobre Manejo de Datos Faltantes

Capítulo 4

METODOLOGÍA

La metodología empleada consiste en 4 principales procesos, los cuales se utilizaron de manera secuencial y son brevemente descritos a continuación. Posteriormente, se explica a detalle cada uno de estos en su respectivo apartado.

1. Proceso de limpieza y filtración:

Se preparan los datos para su posterior análisis, debido que es necesario que se encuentren totalmente limpios, es decir que solo existan valores numéricos y no cadenas de texto ya que se busca extraer información numérica y no categórica. También se filtran delimitando el área de interés que se desea estudiar, en este caso el estado de Sinaloa y los cuerpos de agua que no son subterráneos. Todo esto se realizó empleando hojas de cálculos de Excel las cuales permite manipular rápidamente los datos extraídos de las tablas.

2. Proceso de exploración y análisis de los datos:

Se obtienen estadísticas como los porcentajes de valores faltantes y en base a esto se determinan las variables candidatas para la reconstrucción. También se obtienen sus mecanismos de perdidas, todo esto con la finalidad de seleccionar los métodos de imputación más adecuados para cada una de las variables.

3. Proceso de imputación:

Se aplican las técnicas para el tratamiento de datos faltantes, se realizan imputaciones con distintos métodos y se generan nuevas fuentes de información que sirven para el entrenamiento de los modelos de aprendizaje máquina.

4. Proceso de generación de modelos de aprendizaje máquina:

Se entrenan y evalúan dichos modelos, cada uno de ellos contribuye y ayuda a determinar la siguiente variable.

4.1. Proceso de limpieza y filtración

4.1.1. Panorama general del conjunto de datos

Antes de exponer el proceso de limpieza y filtración se presenta primero el panorama general y la estructura del conjunto de datos utilizado. Esta fuente de información se conforma de las muestras obtenidas a través de los distintos monitores realizado mediante la **RENAMECA** desde 2012 hasta 2020 a nivel federativo, por lo que existen registros de todos los estados y de ahí la necesidad de filtrar solo los pertenecientes al estado de Sinaloa.

La dimensión de dicho conjunto está dada por la siguiente representación:

$$D = (121, 441, 262)$$

A la cual se le denomina shape e indica que se tienen:

- Filas: 121,441
- Columnas: 262

Las 262 columnas se distribuyen de la siguiente manera:

- **CLAVE SITIO** (Un estado puede contener muchos sitios): CADENA DE TEXTO
- **CLAVE MONITOREO** (Identifica el monitoreo): CADENA DE TEXTO
- **NOMBRE DEL SITIO** (Representa el nombre del sitio): CADENA DE TEXTO
- **TIPO CUERPO DE AGUA** (Indica el cuerpo de agua): CADENA DE TEXTO
- **FECHA REALIZACIÓN** (Día, mes y año en que se realizó el monitoreo): FECHA
- **Año** (Contiene el valor numérico del año): NÚMÉRICO

CLAVE PARÁMETRO	PARÁMETRO	UNIDAD MEDIDA
ABS_UV	Absorción UV	U Abs/cm
ACRILONITRILO	Acrilonitrilo	mg/L
ACROLEINA	Acroleína	mg/L
ALC_FEN	Alcalinidad a la Fenoftaleína	mg CaCO3/L
ALC_TOT	Alcalinidad Total	mg CaCO3/L
AS_TOT	Arsénico Total	mg/L
BENCENO	Benceno	mg/L
...	...	...
AS_SOL	Arsénico Soluble	mg/L

- **256 parámetros** (NÚMÉRICO) dados por la tabla de arriba:

Véase el **Anexo A** para consultar la tabla de parámetros completa.

Mientras que en la **figura 4.1** se puede apreciar el panorama general de la base de datos, el conjunto de datos que se utiliza en concreto es el que se deriva de la tabla Monitoreos y es precisamente el que fue descrito anteriormente. Es decir que todas aquellas columnas mencionadas se encuentran en esta tabla y corresponden a la información obtenida en cada monitoreo.

Y por otro lado, las tablas **Sitios** y **Parámetros** son meramente informativas ya que su información no es utilizada para el procesamiento, más sin embargo desempeñan otras funciones. Por ejemplo, en el caso de la primera tabla, es útil para realizar el proceso de filtración, el cual se describe más adelante. Mientras que la segunda como se vio anteriormente permite tener una descripción de cada uno de los parámetros.

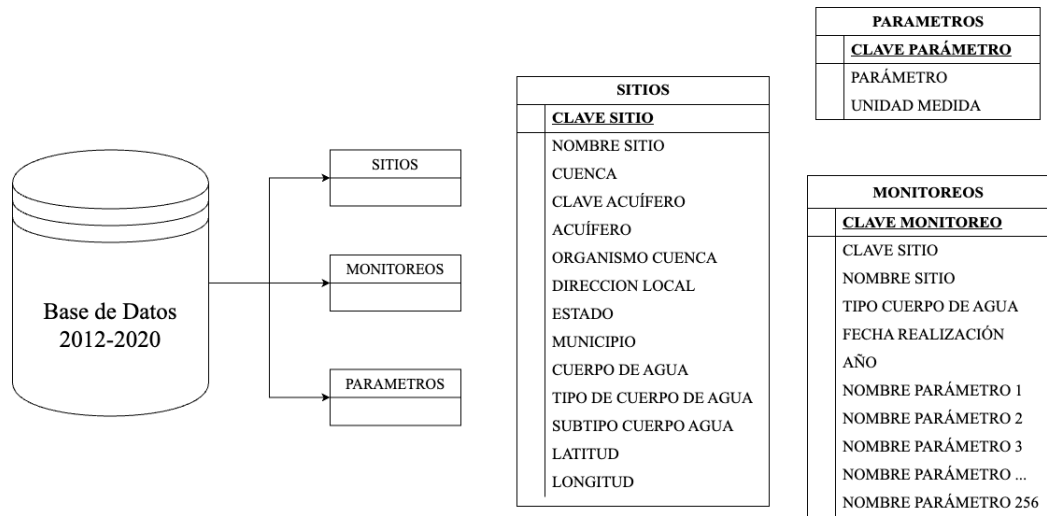


Figura 4.1: Panorama general de la base de datos

4.1.2. Área de interés

El área de interés como se mencionó anteriormente, esta acotada para el estado de Sinaloa y así mismo a aquellos cuerpos de agua que no son subterráneos. De tal manera que las condiciones de filtración quedan dadas por la siguiente expresión:

$$X = \{x \mid \text{CLAVE SITIO} \in \text{SINALOA} \wedge \text{TIPO CUERPO DE AGUA} \neq \text{SUBTERRÁNEO}\} \quad (4.1)$$

Una vez aplicadas estas mismas, se obtuvo un subconjunto del **dataset** original y su **shape** quedó reducido a (6,163, 262).

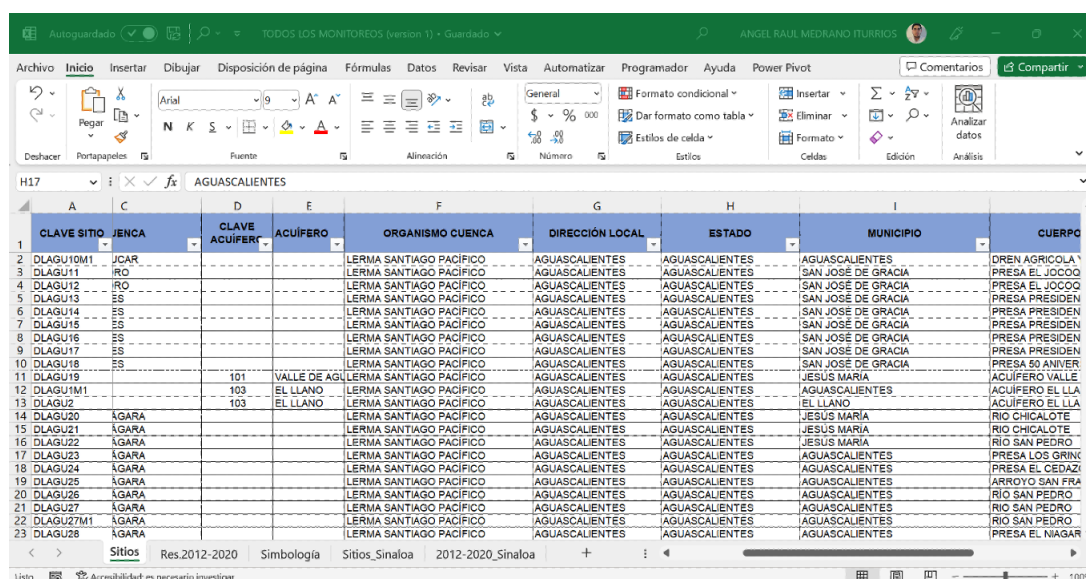
Es decir:

- Filas: 6,163
- Columnas: 262

4.1.3. Proceso de filtración

Como se mencionó anteriormente en el panorama general de la base de datos, la tabla **Sitios** resulta útil para para realizar la filtración para el estado de Sinaloa, ya que en esta tabla se puede encontrar la **CLAVE DEL SITIO** la cual fue empleada para realizar el filtro.

En la **figura 4.2** se muestra que un estado contiene muchos sitios ubicados cada uno en diferentes municipios, así mismo cada sitio se compone por una clave única.



CLAVE SITIO	JENCA	CLAVE ACUIFERO	ACUIFERO	ORGANISMO CUENCA	DIRECCIÓN LOCAL	ESTADO	MUNICIPIO	CUERPO
DLAGU10M1	JCAR			LERMA SANTIAGO PACIFICO	AGUASCALIENTES	AGUASCALIENTES	AGUASCALIENTES	DREN AGRICOLA
DLAGU11	RO			LERMA SANTIAGO PACIFICO	AGUASCALIENTES	AGUASCALIENTES	SAN JOSE DE GRACIA	PRESA EL JOCOO
DLAGU12	RO			LERMA SANTIAGO PACIFICO	AGUASCALIENTES	AGUASCALIENTES	SAN JOSE DE GRACIA	PRESA EL JOCOO
DLAGU13	SS			LERMA SANTIAGO PACIFICO	AGUASCALIENTES	AGUASCALIENTES	SAN JOSE DE GRACIA	PRESA PRESIDEN
DLAGU14	SS			LERMA SANTIAGO PACIFICO	AGUASCALIENTES	AGUASCALIENTES	SAN JOSE DE GRACIA	PRESA PRESIDEN
DLAGU15	SS			LERMA SANTIAGO PACIFICO	AGUASCALIENTES	AGUASCALIENTES	SAN JOSE DE GRACIA	PRESA PRESIDEN
DLAGU16	SS			LERMA SANTIAGO PACIFICO	AGUASCALIENTES	AGUASCALIENTES	SAN JOSE DE GRACIA	PRESA PRESIDEN
DLAGU17	SS			LERMA SANTIAGO PACIFICO	AGUASCALIENTES	AGUASCALIENTES	SAN JOSE DE GRACIA	PRESA PRESIDEN
DLAGU18	SS			LERMA SANTIAGO PACIFICO	AGUASCALIENTES	AGUASCALIENTES	SAN JOSE DE GRACIA	PRESA 50 ANIVER
DLAGU19		101	VALLE DE AGU	LERMA SANTIAGO PACIFICO	AGUASCALIENTES	AGUASCALIENTES	JESUS MARIA	ACUIFERO VALLE
DLAGU1M1		103	EL LLANO	LERMA SANTIAGO PACIFICO	AGUASCALIENTES	AGUASCALIENTES	AGUASCALIENTES	ACUIFERO EL LLA
DLAGU2		103	EL LLANO	LERMA SANTIAGO PACIFICO	AGUASCALIENTES	AGUASCALIENTES	EL LLANO	ACUIFERO EL LLA
DLAGU20	AGARA			LERMA SANTIAGO PACIFICO	AGUASCALIENTES	AGUASCALIENTES	JESUS MARIA	RIO CHICALOTE
DLAGU21	AGARA			LERMA SANTIAGO PACIFICO	AGUASCALIENTES	AGUASCALIENTES	JESUS MARIA	RIO CHICALOTE
DLAGU22	AGARA			LERMA SANTIAGO PACIFICO	AGUASCALIENTES	AGUASCALIENTES	JESUS MARIA	RIO SAN PEDRO
DLAGU23	AGARA			LERMA SANTIAGO PACIFICO	AGUASCALIENTES	AGUASCALIENTES	AGUASCALIENTES	PRESA LOS GRIN
DLAGU24	AGARA			LERMA SANTIAGO PACIFICO	AGUASCALIENTES	AGUASCALIENTES	AGUASCALIENTES	PRESA EL CEDAZ
DLAGU25	AGARA			LERMA SANTIAGO PACIFICO	AGUASCALIENTES	AGUASCALIENTES	AGUASCALIENTES	ARROYO SAN FRA
DLAGU26	AGARA			LERMA SANTIAGO PACIFICO	AGUASCALIENTES	AGUASCALIENTES	AGUASCALIENTES	RIO SAN PEDRO
DLAGU27	AGARA			LERMA SANTIAGO PACIFICO	AGUASCALIENTES	AGUASCALIENTES	AGUASCALIENTES	RIO SAN PEDRO
DLAGU27M1	AGARA			LERMA SANTIAGO PACIFICO	AGUASCALIENTES	AGUASCALIENTES	AGUASCALIENTES	RIO SAN PEDRO
DLAGU28	AGARA			LERMA SANTIAGO PACIFICO	AGUASCALIENTES	AGUASCALIENTES	AGUASCALIENTES	PRESA EL NIAGAR

Figura 4.2: Tabla sitios

Dentro de la columna **ESTADO** se realizó un filtro tal y como se muestra en la **figura 4.3** (véase la siguiente página), seleccionando únicamente al estado de Sinaloa y de esta manera se obtuvieron todas las claves de sitios que se encuentran en dicho estado.

Posteriormente se utilizó la tabla **Monitoreos** y se realizó otro filtro pero ahora en la columna **CLAVE SITIO**, este para las claves que se obtuvieron anteriormente y así de esta manera filtrar los monitoreos para los cuerpos de agua en Sinaloa.

En la **figura 4.5** se aprecia que dichas claves se componen del prefijo “**OCPNO**” y dado que estas mismas son de tipo cadena de texto se aplicó un filtro utilizando dicha condición y así mismo también consideró la clave **MÍNIMA** que corresponde a la **4445** y la **MÁXIMA** que es **6224**, para corroborar que el filtro se realizó correctamente.

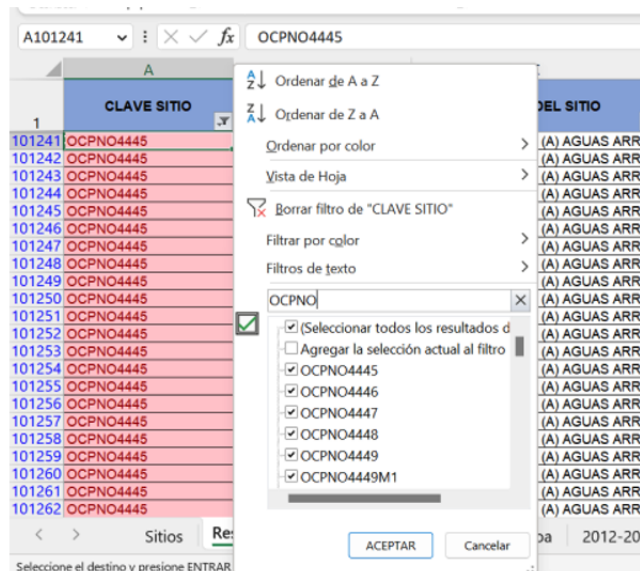


Figura 4.5: Filtro claves Sinaloa

De esta manera se filtraron los monitoreos realizados para el estado de Sinaloa.

	C	D	E	F	G	H	I	J	K	L	M	N	O
	NOMBRE DEL SITIO	TIPO CUERPO DE AGUA	FECHA REALIZACIÓN	Año	ALC_FEN	ALC_TOT	CO3	HCO3	OH	CLOROF	COLL_FEC	COLL_TOT	E_COLI
2	NORTE (A) AGUAS ARRIBA	LÓTICO	15/11/2012	2012							>24196	>24196	
3	NORTE (A) AGUAS ARRIBA	LÓTICO	07/03/2013	2013							>24196	>24196	>24196
4	NORTE (A) AGUAS ARRIBA	LÓTICO	26/04/2013	2013							24196	24196	365
5	NORTE (A) AGUAS ARRIBA	LÓTICO	12/06/2013	2013							>24196	>24196	77
6	NORTE (A) AGUAS ARRIBA	LÓTICO	28/07/2013	2013							663	12997	10
7	NORTE (A) AGUAS ARRIBA	LÓTICO	10/09/2013	2013							>24196	>24196	86
8	NORTE (A) AGUAS ARRIBA	LÓTICO	21/10/2013	2013							24196	>24196	48
9	NORTE (A) AGUAS ARRIBA	LÓTICO	03/03/2014	2014							>24196	>24196	1985
10	NORTE (A) AGUAS ARRIBA	LÓTICO	28/04/2014	2014							>24196	>24196	1985
11	NORTE (A) AGUAS ARRIBA	LÓTICO	06/05/2014	2014							>24196	>24196	61
12	NORTE (A) AGUAS ARRIBA	LÓTICO	22/07/2014	2014							>24196	>24196	517
13	NORTE (A) AGUAS ARRIBA	LÓTICO	08/09/2014	2014							1835	>24196	163
14	NORTE (A) AGUAS ARRIBA	LÓTICO	14/10/2014	2014							>24196	>24196	176
15	NORTE (A) AGUAS ARRIBA	LÓTICO	16/02/2015	2015							24196	>24196	2419
16	NORTE (A) AGUAS ARRIBA	LÓTICO	29/03/2015	2015							>24196	>24196	816
17	NORTE (A) AGUAS ARRIBA	LÓTICO	13/05/2015	2015							>24196	>24196	79
18	NORTE (A) AGUAS ARRIBA	LÓTICO	24/05/2015	2015							>24196	>24196	410
19	NORTE (A) AGUAS ARRIBA	LÓTICO	12/08/2015	2015							>24196	>24196	325
20	NORTE (A) AGUAS ARRIBA	LÓTICO	02/10/2015	2015							24196	>24196	146
21	NORTE (A) AGUAS ARRIBA	LÓTICO	02/03/2016	2016							>24196	>24196	980
22	NORTE (A) AGUAS ARRIBA	LÓTICO	22/04/2016	2016							>24196	>24196	101
23	NORTE (A) AGUAS ARRIBA	LÓTICO	10/06/2016	2016							19863	>24196	55

Figura 4.6: Monitoreos de Sinaloa

El proceso de filtración no concluye aquí ya que tambien es necesario aplicar la segunda condición, en la cual se filtran aquellos monitoreos en los que el cuerpo de agua sea diferente de **SUBTERRÁNEO**.

Para esto se utilizó una macro de **Visual Basic** (véase figura 4.7) para buscar dentro de todas las filas específicamente en la columna “D” la cual corresponde al tipo de cuerpo de agua, aquellas que coincidan con lo mencionado anteriormente y así de esta manera eliminarlas, ya que no son parte del área de interés que se quiere estudiar.

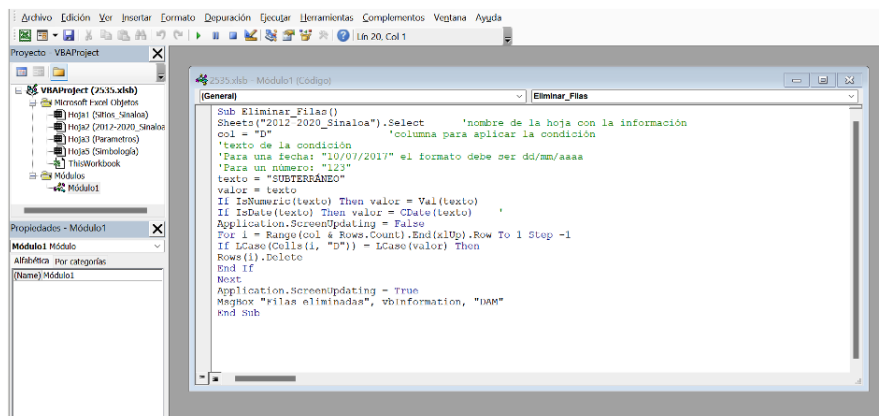


Figura 4.7: Macro de VB para eliminar Subterráneo

4.1.4. Proceso de limpieza

Una vez que se obtuvieron los datos filtrados fue necesario aplicar un proceso de limpieza para remover el carácter “<” ya que como se muestra en la **figura 4.8** existen algunos registros con dicho carácter. Este mismo entorpece el proceso de análisis ya que es necesario que los valores sean totalmente numéricos, y al tener dicho carácter se convierten en cadenas de texto. Así que fue requerido que se eliminaran de manera tal que no se afectara al valor numérico.

B	C	D	E	F	AJ	AK	AL	AM	AN
CLAVE MONITOREO	NOMBRE DEL SITIO	TIPO CUERPO DE AGUA	FECHA REALIZACIÓN	Año	TOX_FIS_FON_5_UT	TOX_V_15_UT	TOX_V_30_UT	TOX_V_5_UT	P_TOT
CPNO4445-080614	CULIACAN NORTE (A) AGUAS ARRIBA/LOTICO		06/06/2014	2014	<1	<1			0.076
CPNO4445-270714	CULIACAN NORTE (A) AGUAS ARRIBA/LOTICO		22/07/2014	2014	<1	<1			0.101
CPNO4445-140914	CULIACAN NORTE (A) AGUAS ARRIBA/LOTICO		08/09/2014	2014	<1	<1			0.145
CPNO4445-261014	CULIACAN NORTE (A) AGUAS ARRIBA/LOTICO		14/10/2014	2014	<1	<1			0.162
CPNO4445-080215	CULIACAN NORTE (A) AGUAS ARRIBA/LOTICO		16/02/2015	2015	<1	<1			0.313
CPNO4445-220315	CULIACAN NORTE (A) AGUAS ARRIBA/LOTICO		29/03/2015	2015	<1	<1			0.2272
CPNO4445-170515	CULIACAN NORTE (A) AGUAS ARRIBA/LOTICO		13/05/2015	2015	<1	<1			0.2064
CPNO4445-280615	CULIACAN NORTE (A) AGUAS ARRIBA/LOTICO		24/06/2015	2015	<1	<1			0.1482
CPNO4445-160815	CULIACAN NORTE (A) AGUAS ARRIBA/LOTICO		12/08/2015	2015	<1	<1			0.274886
CPNO4445-041015	CULIACAN NORTE (A) AGUAS ARRIBA/LOTICO		02/10/2015	2015	<1	<1			0.218002
CPNO4445-280216	CULIACAN NORTE (A) AGUAS ARRIBA/LOTICO		02/03/2016	2016	<1	<1			0.288772
CPNO4445-170416	CULIACAN NORTE (A) AGUAS ARRIBA/LOTICO		22/04/2016	2016	<1	<1			0.390599
CPNO4445-050616	CULIACAN NORTE (A) AGUAS ARRIBA/LOTICO		10/06/2016	2016	<1	<1			0.13754
CPNO4445-170716	CULIACAN NORTE (A) AGUAS ARRIBA/LOTICO		26/07/2016	2016	<1	<1			0.157395
CPNO4445-040916	CULIACAN NORTE (A) AGUAS ARRIBA/LOTICO		12/09/2016	2016	<1	<1			0.158689
CPNO4445-091016	CULIACAN NORTE (A) AGUAS ARRIBA/LOTICO		11/10/2016	2016	<1	<1			0.555031
CPNO4445-190217	CULIACAN NORTE (A) AGUAS ARRIBA/LOTICO		14/02/2017	2017	<1	<1			0.3582
CPNO4445-090417	CULIACAN NORTE (A) AGUAS ARRIBA/LOTICO		09/04/2017	2017	<1	<1			0.20313
CPNO4445-280517	CULIACAN NORTE (A) AGUAS ARRIBA/LOTICO		22/05/2017	2017	<1	<1			0.04157
CPNO4445-090717	CULIACAN NORTE (A) AGUAS ARRIBA/LOTICO		17/07/2017	2017	<1	<1			0.12892
CPNO4445-200817	CULIACAN NORTE (A) AGUAS ARRIBA/LOTICO		22/08/2017	2017	<1	<1			0.36949
CPNO4445-011017	CULIACAN NORTE (A) AGUAS ARRIBA/LOTICO		02/10/2017	2017	<1	<1			

Figura 4.8: Registros con carácter “<”

Para realizar este procedimiento, se seleccionó el rango que corresponde con los 256 parámetros y se utilizó la función buscar y reemplazar para indicarle que se buscara al carácter “<” y lo reemplazara por una cadena vacía en todos aquellos registros que cumplan con la condición, tal y como se aprecia en la **figura 4.9** (véase la siguiente página).

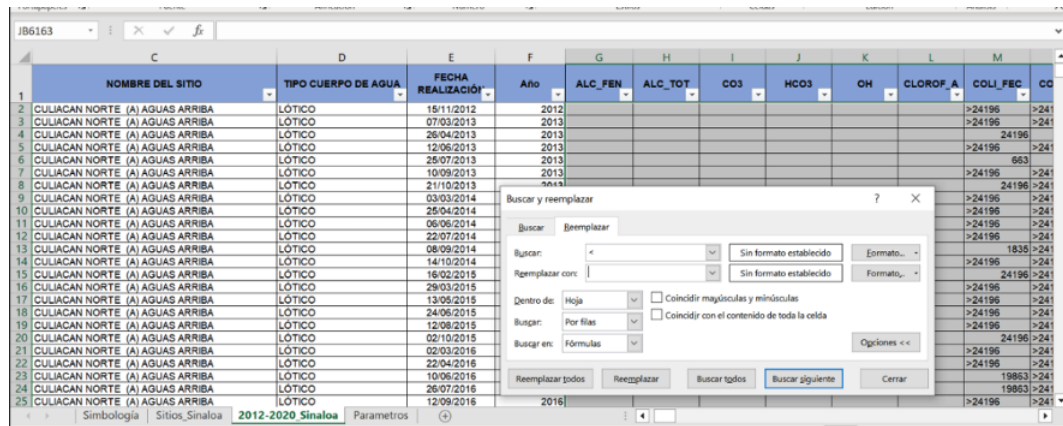


Figura 4.9: Función buscar y reemplazar

4.2. Proceso de exploración y análisis de los datos

Posteriormente después de haber realizado el proceso de limpieza y filtración se pusieron en marcha los procedimientos de exploración para obtener estadísticas de los datos y en base a estos resultados determinar las estrategias para el proceso de imputación.

Primeramente se seleccionaron aquellos parámetros cuyo porcentaje de perdida fuera inferior al 10 % mismos que fueron considerados como posibles candidatos para generar los modelos de imputación. Para esto se obtuvo el total de datos faltantes para cada parámetro y se eligieron solo los que cumplían dicha condición, siendo estos los que se muestran la gráfica que se aprecia en la **figura 4.10**.

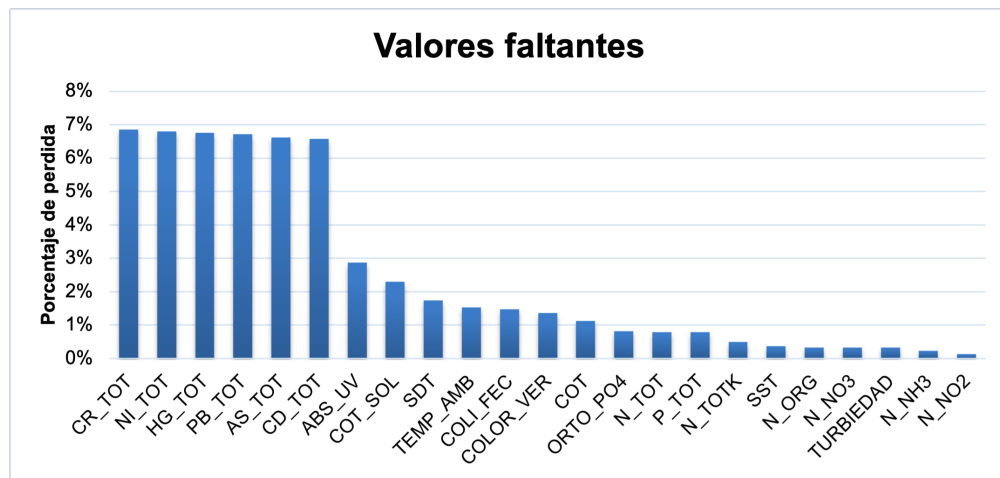


Figura 4.10: Valores faltantes

Mientras que en la **tabla 4.1** se muestra la relación de cada parámetro con su total de valores faltantes, misma que corresponde a la gráfica mencionada anteriormente.

PARÁMETRO	Total
N_NO2	9
N_NH3	15
N_ORG	21
N_NO3	21
TURBIEDAD	21
SST	24
N_TOTK	32
N_TOT	51
P_TOT	51
ORTO_PO4	53
COT	73
COLOR_VER	88
COLI_FEC	96
TEMP_AMB	99
SDT	113
COT_SOL	149
ABS_UV	186
CD_TOT	427
AS_TOT	429
PB_TOT	436
HG_TOT	438
NI_TOT	441
CR_TOT	445

Tabla 4.1: Valores faltantes

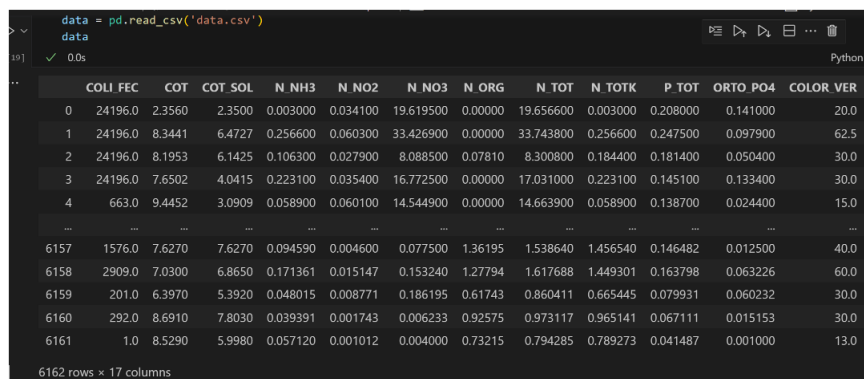
Mas sin embargo, se encontró que para los parámetros **AS_TOT**, **CD_TOT**, **CR_TOT**, **HG_TOT**, **NI_TOT** y **PB_TOT** se tenían valores que se repetían constantemente tal como se aprecia en la **figura 4.11**, lo cual no es favorable ya que solo sesgan los modelos de imputación, por lo que se optó por descartarlos, quedando así un total de 17 parámetros como principales candidatos.

R	S	T	U	V	W	X	Y	Z	AA	AB
ABS_UV	SDT	SST	TURBIEDAD	AS_TOT	CD_TOT	CR_TOT	HG_TOT	NI_TOT	PB_TOT	TEMP_AMB
0.009	768	76.6667	46	0.00139	0.0002	0.0072	0.0006	0.0068	0.00154	35.3
0.0288	433.28	43	60	0.00139	0.0002	0.0112	0.0002	0.0098	0.00154	26.7
0.0086	306.56	45.625	30	0.00139	0.0002	0.00088	0.0002	0.0041	0.00154	34.6
0.009	595.2	54	40	0.028	0.0002	0.0345	0.0001	0.0298	0.00154	
0.0935	748.8	30	5.5	0.00139	0.0002	0.00088	0.0004	0.0002	0.00154	37.4
0.213	581.76	165	62.5	0.00139	0.0002	0.0171	0.0001	0.0154	0.00154	33
0.09	651.52	92	70	0.006	0.0002	0.016	0.001	0.009	0.00154	34.8
0.144	268.16	940	185	0.00139	0.0002	0.0388	0.000459	0.044	0.00154	21
0.135	412.8	226.7	125	0.00139	0.0002	0.015	0.00005	0.012	0.00154	37
0.1	581.12	33	28	0.00139	0.0002	0.0028	0.00028	0.0002	0.00154	38
0.073	751.36			0.0067	0.0002	0.003	0.00009	0.0044	0.00154	34
0.1	705.92	70	30	0.0073	0.0002	0.00088	0.00126	0.008	0.00154	35
0.103	879.36	86	48	0.00139	0.0002	0.00088	0.00009	0.0002	0.00154	34
0.169	416.64	16.67	160	0.00139	0.0002	0.00088	0.000197	0.0191	0.00154	23.5
0.142	361.6	160	125	0.00139	0.0002	0.0103	0.000075	0.0114	0.00154	34
0.086	481.92	126	95	0.012	0.0002	0.00088	0.000056	0.0002	0.00154	36
0.082	718.08	52.5	32	0.00139	0.0002	0.00088	0.000098	0.0002	0.00154	36
0.073	750.08	195	105	0.00139	0.0002	0.0091	0.000204	0.0117	0.00154	38
0.12395	631.04	110	75	0.00139	0.0002	0.008	0.000198	0.0094	0.00154	36
0.13113	371.2	156	28	0.00139	0.0002	0.0192	0.000078	0.0179	0.00154	31
0.15519	292.48	125	60	0.00139	0.0002	0.0178	0.00009	0.0151	0.00154	32
0.11163	424.96	72.86	29	0.00139	0.0002	0.0134	0.000132	0.0108	0.00154	29

Figura 4.11: Valores repetitivos

4.2.1. Identificando mecanismos de perdidas

Para determinar los mecanismos de pérdidas de los datos, primero se cargaron mediante la función **read_csv()** de la librería **pandas**, la cual permite leer los datos en este formato y mantenerlos en estructuras denominadas **DataFrame**, mismos que permanecen en memoria durante la ejecución del programa y pueden ser exportados después de haberlos manipulado.



```
data = pd.read_csv('data.csv')
data
```

	COLI FEC	COT	COT SOL	N NH3	N NO2	N NO3	N ORG	N TOT	N TOTK	P TOT	ORTO PO4	COLOR VER
0	24196.0	2.3560	2.3500	0.003000	0.034100	19.619500	0.000000	19.656600	0.003000	0.208000	0.141000	20.0
1	24196.0	8.3441	6.4727	0.256600	0.060300	33.426900	0.000000	33.743800	0.256600	0.247500	0.097900	62.5
2	24196.0	8.1953	6.1425	0.106300	0.027900	8.088500	0.07810	8.300800	0.184400	0.181400	0.050400	30.0
3	24196.0	7.6502	4.0415	0.223100	0.035400	16.772500	0.000000	17.031000	0.223100	0.145100	0.133400	30.0
4	663.0	9.4452	3.0909	0.058900	0.060100	14.544900	0.000000	14.663900	0.058900	0.138700	0.024400	15.0
...	...	...	...	...	...	...	...	...	...	...	...	...
6157	1576.0	7.6270	7.6270	0.094590	0.004600	0.077500	1.36195	1.538640	1.456540	0.146482	0.012500	40.0
6158	2909.0	7.0300	6.8650	0.171361	0.015147	0.153240	1.27794	1.617688	1.449301	0.163798	0.063226	60.0
6159	201.0	6.3970	5.3920	0.048015	0.008771	0.186195	0.61743	0.860411	0.665445	0.079931	0.060232	30.0
6160	292.0	8.6910	7.8030	0.039391	0.001743	0.006233	0.92575	0.973117	0.965141	0.067111	0.015153	30.0
6161	1.0	8.5290	5.9980	0.057120	0.001012	0.004000	0.73215	0.794285	0.789273	0.041487	0.001000	13.0

6162 rows x 17 columns

Acto seguido, se utilizó la función **matrix()** de la librería **missingno** la cual es útil para la visualización de valores faltantes, para poder generar la gráfica que se muestra en la **figura 4.12**. Mismo en la que se observa la distribución de los valores faltantes para cada uno de los parámetros mediante las líneas blancas.

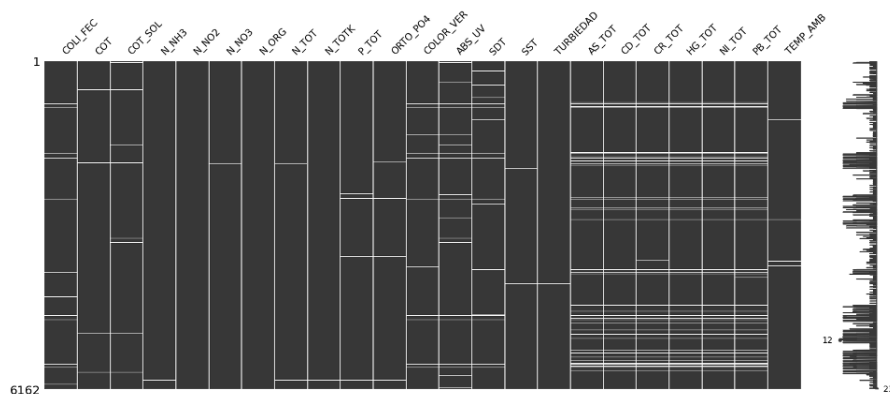


Figura 4.12: Mecanismos de perdidas

Al analizar los mecanismos de pérdidas se detectó que los parámetros que fueron descartados anteriormente seguían un patrón de tipo **MAR**, puesto que en todos se observa que su presencia de valores faltantes están ligados los uno a los otros. Así mismo dichos parámetros son los que tiene una mayor cantidad de datos faltantes por lo que el patrón revela que es debido a la dependencia que existe entre ellos.

Con el fin de seguir analizando e identificando mecanismos de perdidas en los datos, se empleó otro gráfico que también aporta una visión sobre el comportamiento de los datos en base a la ausencia de sus valores. Así mismo se generó con la librería mencionada anteriormente, dicho grafico es una matriz que muestra la correlación entre las variables y sus valores faltantes, tal como se aprecia en la **figura 4.13**.

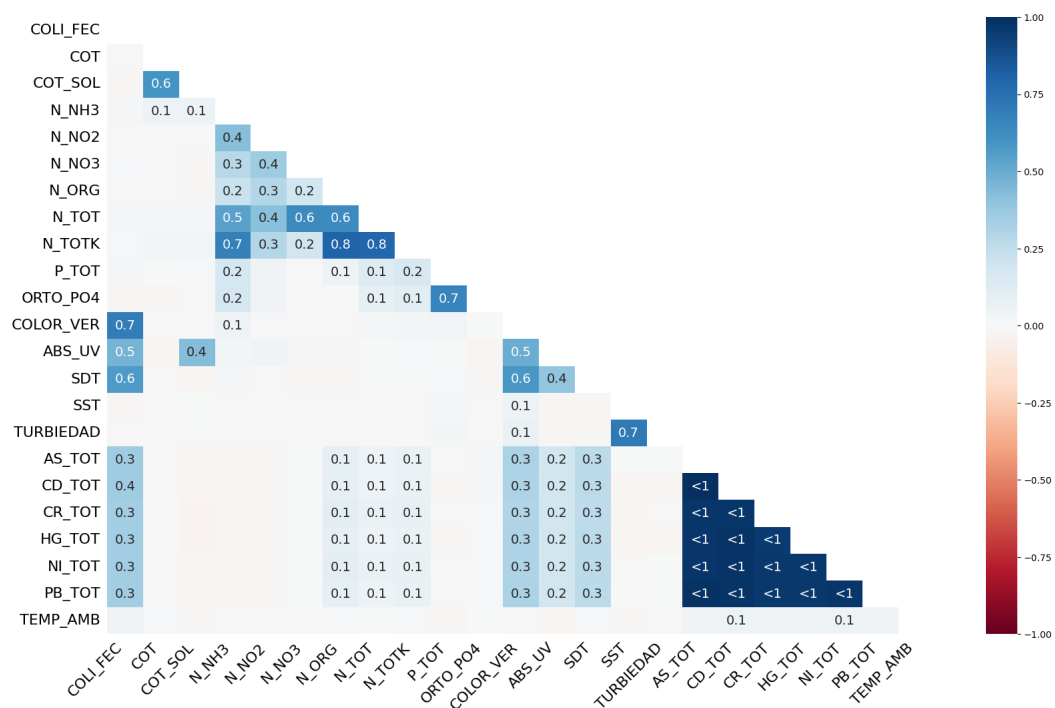


Figura 4.13: Matriz de correlación de pérdida

Cuando se tiene un grado de correlación en **1** indica que si el valor de una variable está presente, entonces el valor de la otra definitivamente está presente.

Mientras que cuando se tiene correlación negativa, es decir en **-1** indica que si el valor de una variable está presente, el valor de las otras variables está definitivamente ausente, tal como se aprecia en la **figura 4.14**.

O	P	Q	R	S	T	U	V	W	X
SST	TURBIEDAD	AS_TOT	CD_TOT	CR_TOT	HG_TOT	NI_TOT	PB_TOT	TEMP_AMB	
12	3.4	0.0018	0.00023	0.00099	0.0001	0.00036	0.00154	27	
10	3.4	0.0018	0.00023		0.0001	0.065	0.00154	28	
10	3.5	0.0018	0.00023	0.00099	0.0001	0.00036	0.00154	31	
54.5	3.8	0.0018	0.00023	0.00099	0.00036	0.00036	0.00154	34	
31	4	0.0018	0.00023	0.00099	0.0001	0.00036	0.00154	30	
22	1.6	0.0018	0.00023	0.00099	0.0001	0.00036	0.00154	30	
27.5	2.3	0.0018	0.00023	0.00099	0.0001	0.0026	0.00154	32	
27.5	3.6	0.0018	0.00023	0.00099	0.0001	0.00036	0.00154	28	
26.25	2.4							29.3	
17	3							30	
37.25	17							29.5	
28	2.5							30.4	
9.5	3							34.6	
11.5	4							34.7	
8	4							35.2	
9.2	1.5	0.00139	0.0002	0.00088	0.0003	0.0002	0.00154	25	
18.8	3.5	0.00139	0.0002	0.00088	0.00036	0.0002	0.00154	26	
17.1	2.5	0.00139	0.0002	0.00088	0.00091	0.0002	0.00154	34	
22	0.78	0.00139	0.0002	0.00088	0.00033	0.0002	0.00154	34	
16	2.6	0.0018	0.00023	0.00099	0.000073	0.00036	0.00154	30	
21	4.5	0.0018	0.00023	0.00099	0.0001	0.00036	0.00154	27	
24.44	5.8	0.0018	0.00023	0.00099	0.0001	0.00036	0.00154	35	

Figura 4.14: Presencia de valores MAR

En la **tabla 4.2** (véase la siguiente hoja) se expresan los mecanismos de pérdidas de cada uno de los parámetros y se da una justificación a esto. Si bien las gráficas mencionadas ayudan a determinar estos mismo, otra estrategia que se utilizó fue revisar los datos a través de hojas de cálculo y se rellenaron los valores faltantes de cada parámetro con un carácter, esto con la intención de después aplicar un formato condicional para resaltar estos valores. Asi pues se realizó el análisis apoyándose de las gráficas y verificando el comportamiento de los datos, de tal manera que fue posible determinar los mecanismos de perdidas.

Otro gráfico útil también es el de la matriz de correlación lineal entre los parámetros, ya que como se verá en la tabla mencionada resulto útil para determinar los mecanismos de pérdidas de tipo MCAR ya que estos no se relacionan con ninguna variable. Dicho gráfico se encuentra en el **Anexo B** mismo que también fue empleado en el proceso de imputación.

Parámetros	Mecanismo	Justificación
COT y COT_SOL	MNAR	Si bien en ocasiones faltan datos exactamente en ambos parámetros, hay casos en los que solo se presenta la ausencia en uno de ellos. Por lo que no son MCAR debido a que existe una correlación entre estos ni tampoco se pueden considerar MAR .
COLI_FEC	MCAR	No existe correlación entre las demás variables donde coinciden con valores faltantes, por lo cual no hay manera de explicar su pérdida a través de alguna de ellas.
N_NH3	MAR	Cada dato faltante en N_NH3 falta de la misma manera en N_TOT y N_TOTK .
N_NO2	MAR	Cada dato faltante en N_NO2 falta de la misma manera en N_TOT .
N_NO3	MAR	Cada dato faltante en N_NO3 también falta en N_TOT .
N_ORG	MAR	Cada dato faltante en N_ORG también falta en N_TOT y N_TOTK .
N_TOT	MNAR	Sus pérdidas aparecen con otros parámetros pero no se tiene alguno que pueda describir esta ausencia de información.
N_TOTK	MAR	Cada dato faltante en N_TOTK también falta en N_TOT .
P_TOT y ORTO_PO4	MNAR	Hay casos en los que faltan en ambos parámetros, así mismo tienen pérdidas de información de manera separada.
COLOR_VER, ABS_UV y SDT	MNAR	En algunas ocasiones aparece en COLOR_VER , ABS_UV y SDT pero esto no siempre se da, por lo que no existe tal dependencia.
SST y TURBIEDAD	MNAR	Aparecen con valores faltantes de manera simultánea, más sin embargo en otras ocasiones es de manera individual, así mismo se tiene una correlación lineal entre ellos.
TEMP_AMB	MCAR	Mismo caso que COLI_FEC .

Tabla 4.2: Mecanismos de pérdidas

4.3. Proceso de imputación

Antes de realizar el proceso de imputación, es importante contar con datos totalmente limpios y filtrados. También haber realizado un análisis para determinar aquellas variables candidatas, los porcentajes de pérdidas de cada una de ellas y los mecanismos de pérdida son útiles para dictaminar esto.

Una vez aplicados estos procedimientos, también se requiere plantear una estrategia para evaluar el desempeño de los modelos de imputación y así mismo comprobar la veracidad de los datos imputados, ya que estos serán útiles para la generación de modelos de aprendizaje máquina, ahí la importancia de realizar con cautela este proceso.

Una estrategia fue aplicar un **análisis de caso de estudios completo**, eliminando todas aquellas filas donde se tenga algún valor faltante. Esto con la intención de tener un conjunto de datos completos y eliminar de manera aleatoria el mismo porcentaje que se tiene originalmente en cada variable y así comparar los valores imputados con los que resultan del análisis.

4.3.1. Análisis de caso de estudios completo

Para realizar un análisis de datos completos se eliminaron primeramente todos los registros donde se encontraban valores faltantes en forma de lista, tal cual lo indica el método de **list-wise deletion**.

Este proceso se realizó con la librería **pandas** mediante la función **drop.na()** la cual elimina todos aquellos registros donde se tiene información ausente. Así entonces se obtuvo un conjunto de datos al que denominaremos **data_set_nNan** como manera representativa para indicar que es información que no tiene valores ausentes, es decir que se encuentra completo.

Posteriormente aplicado esto, se eliminó de manera aleatoria la misma cantidad de valores ausentes que se tenían en cada parámetro y se generó otro conjunto de datos con información eliminada aleatoriamente al cual le nombraremos simplemente como **data_set**. De tal manera que con el conjunto de datos completos y el que se menciono fue posible evaluar los resultados obtenidos a través de las distintas técnicas de imputación utilizando las métricas de evaluación **MSE**, **RMSE** y **R2**.

La metodología empleada para generar los modelos de imputación se describe a través de la figura 4.18.

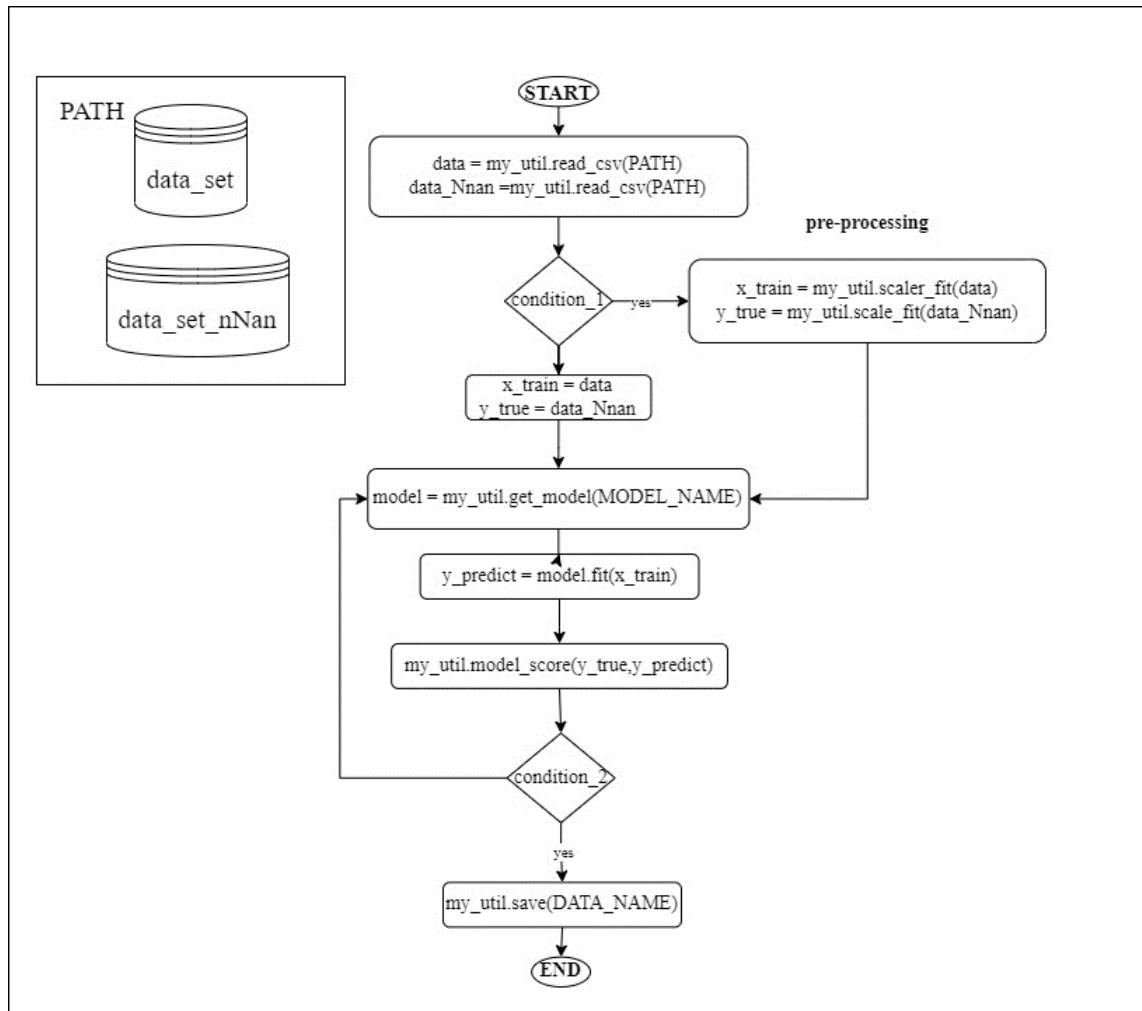


Figura 4.15: Metodología para generar modelos de imputación

Dicha figura ilustra el flujo de trabajo que se siguió para generar los modelos de imputación. Cabe mencionar que se construyó una biblioteca donde se agruparon todas las funciones necesarias que se emplearon para este proceso, la cual permitió tener una mejor estructura. A continuación se detalla cada flujo del diagrama.

Paso 1:

Se cargan los datos a través de la librería **pandas** en un **DataFrame**, esto se realiza mediante la librería propia construida para reutilizar y simplificar el flujo de trabajo. Se utilizaron dos conjuntos de datos, el primero corresponde a los parámetros que se desean imputar y corresponde con el denominado **data_set** al cual se le realizaron eliminaciones aleatorias tal como fue descrito anteriormente. El segundo corresponde el denominado **data_set_nNan** el cual contiene información completa y certera que será útil para evaluar los resultados.

Paso 2:

Se evalúa la primera condición en la cual se decide si los datos necesitan algún preprocesamiento como lo es escalarlos en el mismo rango, esto con la finalidad de que todos los valores estén en la misma distribución y el modelo sea menos propenso a errores. En algunos casos no es necesario aplicar este preprocesamiento dado que si se utilizan variables que tengan la misma distribución este procedimiento no causa ningún efecto y se dejan los datos tal cual como están. Pero por el contrario, si se trabajan variables con distribuciones distintas lo más ideal es aplicarles una normalización, dicho proceso se realiza a través de la función **MinMaxScaler()** de **scikit-learn** la cual normaliza los valores en un rango de 0 a 1, donde el valor de 1 corresponde con el más grande de toda la distribución. Así entonces los datos son almacenados dentro de **arrays** de **numpy** en las variables **x_train** y **y_true**.

Paso 3:

El procedimiento anterior corresponde a lo que se le denomina preparar los datos para el entrenamiento o procesamiento y como se vio consiste en aplicarles transformaciones a los datos y separarlos mediante estructuras para posteriormente procesarlas a través de alguna técnica. Las técnicas empleadas para imputar los datos fueron las siguientes:

- IterativeImputer (**MICE**) y KNN, SimpleFill a través de la librería **fancyimpute**
- IterativeImputer, KNNImputer, SimpleImputer a través de la librería **scikit-learn**.

Así entonces se selecciona algunas de las técnicas mencionadas a través de la función **get_model()** indicándole el nombre de la misma y se almacena en la variable **model**, a la cual posteriormente se le pide que realice el proceso de imputación mediante la función **fit()** enviando como parámetro la estructura que contiene los datos que se desean imputar y el resultado es almacenado en la variable **y_predict**.

Paso 4:

Por último, una vez que se obtuvieron los datos imputados se realiza una evaluación. A través de la función **model_score()** se envían tanto como los datos imputados y los que originales. Así entonces la función se encarga de aplicar las métricas de evaluación utilizando a su el módulo **metrics** de la librería de **scikit-learn** y regresa por consola el resultado de estas, en base a esto se acciona la segunda condición en la que si las métricas cumplen con los criterios establecidos se procede a guardar la información imputada, caso contrario se selecciona alguna otra métrica y así el proceso continua hasta

obtener un resultado favorable.

De no obtener buenos resultados al probar distintas métricas es un indicativo de que la técnica empleada no es la más adecuada o que la información que se está procesando no aporta mucho, por lo que se puede optar por elegir otros parámetros. Esta condición ayuda a experimentar cómo se comportan ciertas variables al aplicarle las distintas técnicas, es meramente ilustrativa ya que la decisión la debe de tomar el usuario que se encuentre realizando las imputaciones, comparando los resultados de las métricas y revisando los datos. Así mismo otra comprobación después de obtener los datos imputados es revisar su distribución, la cual no debe de presentar cambios después de aplicar este proceso, tal como se aprecia en la figura 4.18

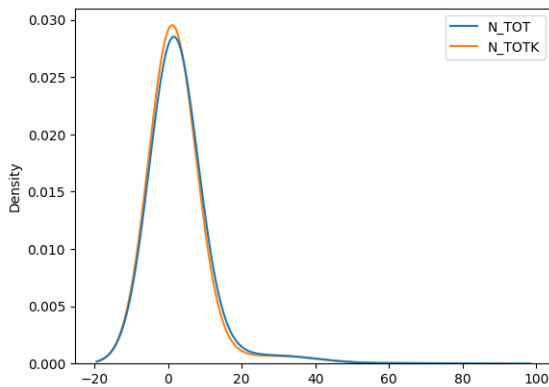


Figura 4.16: Distribución de los datos

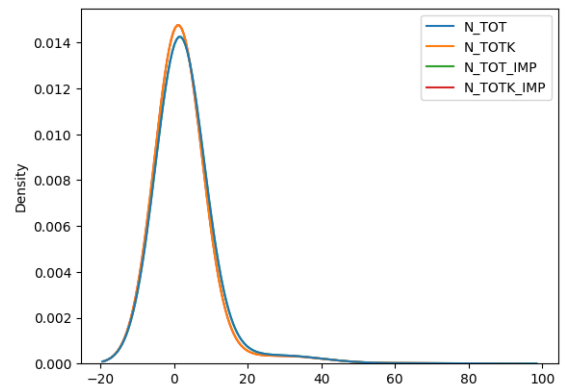


Figura 4.17: Distribución de los datos después de imputar

Figura 4.18: Comparación de la distribución de los datos

4.4. Proceso de generación de modelos de aprendizaje máquina

De la misma manera que en el proceso anterior, la metodología empleada para la generación de modelos de aprendizaje máquina se describe a través del siguiente diagrama ilustrado en la figura **ml_models**.

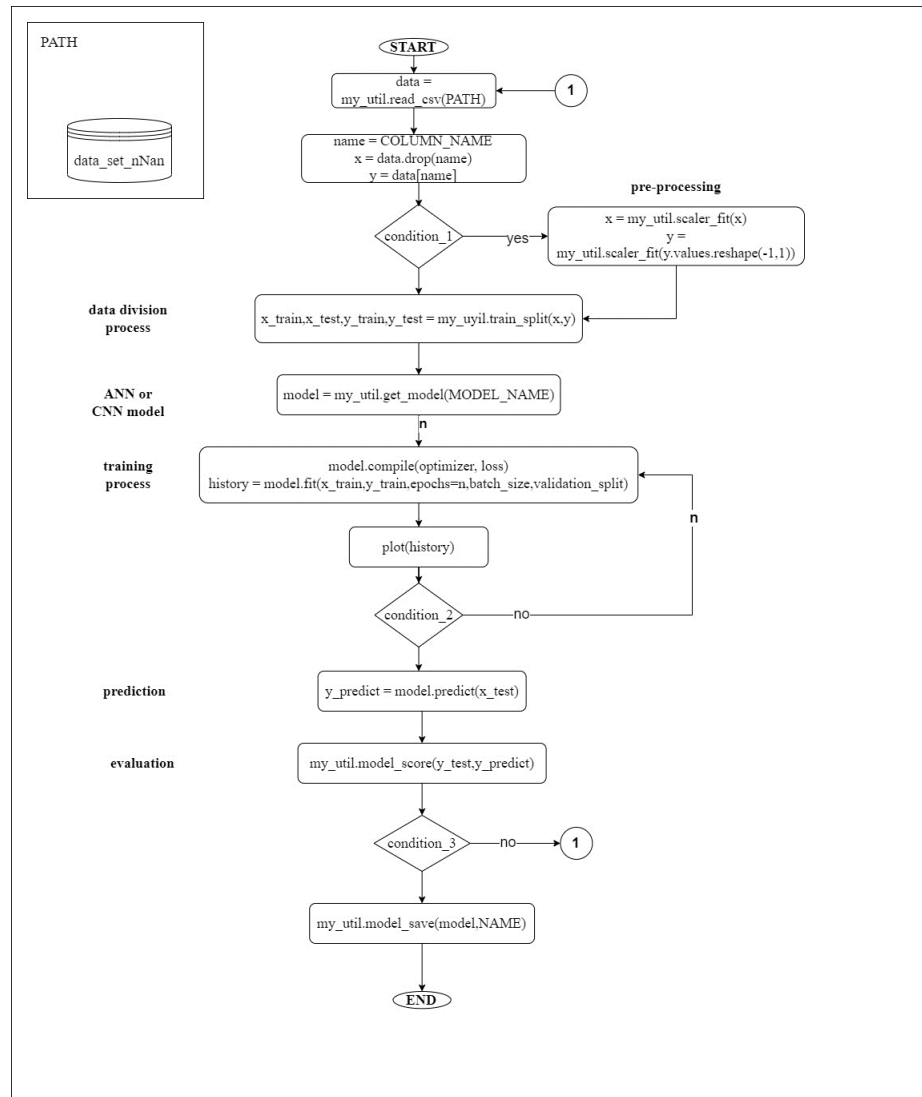


Figura 4.19: Metodología para generar modelos de aprendizaje máquina

Paso 1:

De la misma manera que en proceso anterior cargan los datos a través de la librería **pandas** en un **DataFrame**, esto se realiza mediante la librería propia construida para reutilizar y simplificar el flujo de trabajo. Posteriormente se define la variable que se quiere predecir y se asigna en la variable name la cual se utiliza para indicarle a dicha estructura que elimine la columna que coincida con ese nombre y el resultado lo almacene en la variable x la cual está destinada para los datos de entrenamiento, de la misma manera se le indica al DataFrame que asigne en la variable y el contenido de la columna que se eliminó anteriormente, dicha variable se destina para pruebas.

Paso 2:

Como se explicó en la metodología anterior, durante el preprocesamiento se debe realizar cuando los datos no se encuentran en la misma distribución, en este caso como se están utilizando todas las variables si es necesario aplicar un proceso de escalamiento, ya que contribuye al procesamiento de la red. Solo en caso de utilizar un conjunto de datos cuya distribución se encuentra en el mismo rango no es necesario aplicar tal procesamiento. Posteriormente como parte de este, también se divide el conjunto de datos para entrenamiento y pruebas, en este caso se utiliza un margen de **0.3** lo que indica un 70-30 respectivamente.

Paso 3:

Después de dividir el conjunto de datos, se selecciona un modelo de aprendizaje máquina, en este caso se implementaron 3: dos **ANN** y una **CNN**. Después de obtener el modelo a través de la librería propia, se compila el modelo utilizando el optimizador

adam y el **MSE** para la función de pérdida (**loss**). Acto seguido se procede a entrar dicho modelo asignando el número de épocas (**n**) el **batch_size** y **validation_split** cuyos valores por defecto son 32 y 0.2 respectivamente, el entrenamiento concluye una vez han pasado el total de épocas.

Paso 4:

Al terminar el entrenamiento se procede a visualizar de manera gráfica el comportamiento del modelo durante su proceso de aprendizaje, en base a esto se toma la decisión de volver a entrenarlo o no. La grafica revela la relación entre las épocas y la función de pérdida expresada por el Paso 4: Al terminar el entrenamiento se procede a visualizar de manera gráfica el comportamiento del modelo durante su proceso de aprendizaje, en base a esto se toma la decisión de volver a entrenarlo o no. La grafica revela la relación entre las épocas y la función de pérdida expresada por el MSE de tal manera que se puede ir ajustando las épocas si los resultados de la métrica no son favorables. de tal manera que se puede ir ajustando las épocas si los resultados de la métrica no son favorables.

Paso 5:

Acto seguido, se procede a realizar una predicción y así mismo se evalúa que tan buena fue esta misma a través de la función **model_score()** la cual realiza los cálculos de para obtener las respectivas métricas. De obtener resultados favorables se procede a guardar el modelo para poder después reconstruir la información a través de este mismo, caso contrario se toma la estrategia de utilizar otro conjunto de datos, ya sea agregando o quitando columnas.

Capítulo 5

RESULTADOS

5.1. Resultados obtenidos a través de los métodos de imputación

Las estrategias aplicadas en cada parámetro varia, dado a los mecanismos y a su correlación con las demás variables. Se aplico tanto como imputación simple y múltiple tal como se presenta a continuación.

COLI_FEC

Para llevar a cabo el proceso de imputación sobre dicho parámetro, se optó por utilizar imputación simple. Dado que esta variable no compartía mucha correlación con las demás (véase Anexo B). Por otro lado la distribución de estos datos se encontraba en su mayoría predominando con ciertos valores los cuales se presentaban constantemente, por lo que aplicar una imputación simple por la estrategia del valor más frecuente a través de la librería **skit-learn** la cual dio buenos resultados. Así mismo en la **tabla 5.1** se puede apreciar la diferencia de cuando se utiliza imputación múltiple a través de los métodos iterativos.

skit-learn		
Método	R2	MSE
SimpleImputer by Mean	0.9999246079758297	1.416537006609697e-07
SimpleImputer by Median	0.9999880430261309	2.2465898958337353e-08
SimpleImputer by Most frequent	0.9999855263329442	2.719450131699933e-08
IterativeImputer	0.9999246079758297	1.416537006609697e-07

Tabla 5.1: Resultados al imputar COLI_FEC

COT y COT_SOL

Dado que dichas variables presentaban un fuerte correlación la mejor estrategia fue emplear imputación múltiple. Obteniendo así el resultado más favorable como se muestra en la **tabla 5.2** (véase la siguiente hoja), mediante **KNN Imputer** a través de **scikit-learn** y bajo la estrategia de pesos uniformes con $K=9$.

KNN Imputer	scikit-learn			
N_NEIGHBORS	R2_UNI	MSE_UNI	R2_DIS	MSE_DIS
3	0.9987544345254258	0.05353058524155954	0.9987099487977683	0.05418383762095133
5	0.9990529275941674	0.041433175101944546	0.9989117048887085	0.04635053371184862
7	0.9992388783480417	0.03227412779185857	0.9990199899915133	0.04163323914107048
9	0.9992804216307638	0.02899700554203891	0.9990747465243388	0.03876615040989945

Tabla 5.2: Resultados al imputar COT y COT_SOL

N_NH3, N_TOT, N_TOTK, P_TOT, ORTO_PO4

Dado que dichos parámetros presentaban también fuertes niveles de correlación, se optó por emplear imputación múltiple, dando un resultado favorable utilizando imputación iterativa a través de la librería **fancyimpute** tal como se muestra en la **tabla 5.3**.

fancyimpute		
Método	R2	MSE
IterativeImputer	0.999720138587558	0.0018539103675461962
SimpleFill	0.9970281484430515	0.07978345877662127

Tabla 5.3: Resultados al imputar N_NH3, N_TOT, N_TOTK, P_TOT, ORTO_PO4

N_NO2

En el caso de **N_NO2** se utilizó **N_TOT** dado que el mecanismo de pérdida se pudo explicar a través de esta variable, también se incluyó **P_TOT** ya que presenta buena correlación con el parámetro anterior, de manera que se utilizaron estos parámetros que ya habían sido imputados anteriormente. Lo cual resultó favorable a la hora de aplicar el procedimiento de imputación múltiple, siendo el método de imputación iterativa el que mejores resultados brindó, tal como se muestra en la **tabla 5.4**.

fancyimpute		
Método	R2	MSE
IterativeImputer	0.9996663981228524	0.0058705787400870025
SimpleFill	0.9972290286071441	0.0816350525803568

Tabla 5.4: Resultados al imputar N_NON2

N_ORG

De la misma manera se utilizaron datos ya imputados en este caso **N_TOTK** dado que como en el caso anterior se presentó que para los casos donde ocurrían valores faltantes en **N_ORG** se tenían también en dicha variable, adicionalmente se agregaron los parámetros **N_TOT** y **P_TOT** los cuales contribuyeron dada su fuerte correlación con estos. De manera que los resultados obtenidos se muestran en la **tabla 5.5**, donde se puede apreciar que el mejor resultado fue obtenido a través del método de **KNN** de la librería **fancyimpute** con **K=9**.

KNN	fancyimpute	
n_neighbors	R2	MSE
3	0.9998998118392576	0.0010278436539091478
5	0.9999065357669844	0.0009979168721366654
7	0.999910695572839	0.0010092507499852924
9	0.9999154937658514	0.0009312421266034513

Tabla 5.5: Resultados al imputar N_ORG

De esta manera se lograron imputar 10 parámetros de los 17 que habían sido seleccionados como posibles candidatos y como se vio se obtuvieron resultados favorables. Gracias a este proceso fue posible generar modelos de aprendizaje maquina utilizando redes neuronales de tipo **ANN** y **CNN** las cuales permitieron reconstruir un total de 23 parámetros, además de los ya mencionados se incluyeron otros más cuyo porcentaje de perdida oscilaba entre un 40 y 50 %.

5.2. Resultados obtenidos a través de los modelos de aprendizaje máquina

5.2.1. Resultados obtenidos a través de CNN

Dado a que se obtuvieron 10 parámetros reconstruidos a través de las técnicas de imputación, se utilizó ese conjunto de datos para generar primeramente un modelo con CNN para los mismo parámetros, obteniendo los resultados que se muestran en la **tabla 5.8**, de la que se puede rescatar que solamente para **COLI_FEC** no se obtuvieron mejores resultados.

CNN	Métricas		Imputación
Y	R2	MSE	MSE
COLI_FEC	−0,007559942899116434	0.0.0009075057741674788	2.719450131699933e-08
COT	0.8743923551471765	0.00042680971387674575	0.02899700554203891
COT_SOL	0.8540695269886966	0.00048818525189160493	0.02899700554203891
N_NO2	−0,5392095407249384	0.00037329405689141804	0.0058705787400870025
N_NH3	0.9995998771847171	2.5759440079486577e-06	0.0018539103675461962
N_ORG	0.9950689238164462	7.2059646237581405e-06	0.0009312421266034513
N_TOT	0.9218208107899076	0.0005063388820642076	0.0018539103675461962
N_TOTK	0.9988122966547189	7.93320309384626e-06	0.0018539103675461962
P_TOT	0.8888363934339495	0.0003143762492801728	0.0018539103675461962
ORTO_PO4	0.8810407220625038	0.0003364227985248641	0.0018539103675461962

Tabla 5.6: Resultados obtenidos CNN (10)

N_NO3

Utilizando los 10 parámetros anteriores se entrenó un modelo para predecir el parámetro **N_NO3**, dicho entrenamiento se realizó con 50 épocas, obteniendo el resultado que se aprecia en la **figura 5.1**.

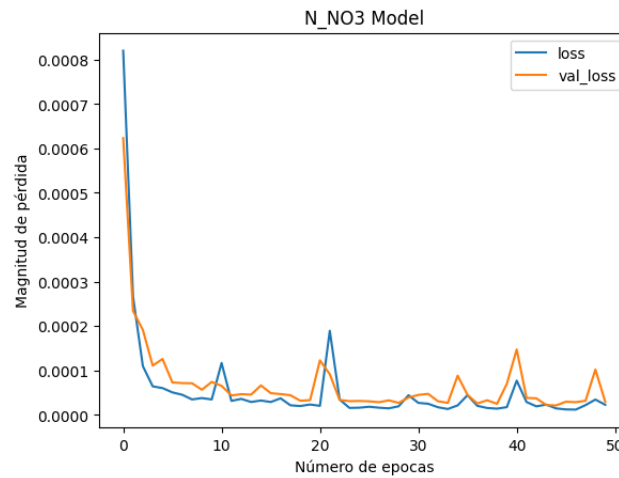


Figura 5.1: Resultados obtenidos para N_NO3 con CNN

Posteriormente se evaluó el resultado de la predicción y se obtuvo que se muestra en la **tabla 5.7**.

CNN	Métricas	
Y	R2	MSE
N_NO3	0.9901924760288694	0.005590559728887884

Tabla 5.7: Resultados obtenidos para N_NO3 con CNN

SDT, COLOR_VER, ABS_UV, SST, TURBIEDAD, TEMP_AMB

Para generar dichos parámetros los cuales no fueron imputados se utilizaron los 11 anteriores los cuales fueron pronosticados a través de la **CNN**, obteniendo los resultados que se muestran en la **tabla 4-8**, misma en la que se aprecia que los resultados de la métrica **R2** se mantuvieron más alejados que en los anteriores y adicionalmente se agregó la métrica **RMSE**. También se nota como en la **TEMP_AMB** no se obtuvieron resultados tan favorables.

CNN	Métricas		
Y	R2	MSE	RMSE
SDT	0.7706196114807364	0.008346128665616619	0.09135714895735647
COLOR_VER	0.5011689497594578	0.001177853166303535	0.02899700554203891
ABS_UV	0.5155614856008873	0.004097146530654881	0.064008956644011
SST	0.6194312276238687	0.0007218030829506385	0.026866393188342913
TURBIEDAD	0.7283041969155735	0.000708281857145111	0.026613565284364118
TEMP_AMB	0.13029098340274148	0.012643434926628769	0.11244302969339082

Tabla 5.8: Resultados obtenidos para SDT, COLOR_VER, ABS_UV, SST, TURBIEDAD y TEMP_AMB con CNN

5.2.2. Resultados obtenidos a través de ANN

Los resultados obtenidos a través del primer modelo de ANN se generaron utilizando todas las variables disponibles en este caso 16 para cada uno de los parámetros respectivamente, dichos resultados se muestran en la **tabla 5.9** y estos solo muestran mejoras en los parámetros **N_NO3** y **N_TOT** en comparación a los resultados anteriores, pero se puede atribuir que esto es debido al mecanismo de perdida de **N_NO3** el cual como se vio es de tipo **MAR** y es totalmente dependiente de **N_TOT**.

ANN	Métricas		
Y	R2	MSE	RMSE
COLI_FEC	0.09257162971099175	0.0008173176112094741	0.0285887672208767
COT	0.8790907591140487	0.00041084472659330657	0.0202693050347886
COT_SOL	0.8553812351239681	0.00048379715834807093	0.021995389479344776
N_NH3	0.997819350988004	1.403876395280827e-05	0.0037468338571129984
N_NO2	-0,22712795381596051	0.00029760702495975814	0.01725129053026927
N_NO3	0.9760356754008108	7.63688760208354e-05	0.008738928768495335
N_ORG	0.9757544151315972	3.54309729439469e-05	0.005952392203471383
N_TOT	0.9993271342345973	4.357912928947191e-06	0.002087561479082039
N_TOTK	0.9977246864131795	1.5197839475785276e-05	0.003898440646692633
P_TOT	0.945861098327577	0.0001531075265879536	0.012373662618156096
ORTO_PO4	0.897574129203052	0.0001875976485202529	0.013696629093329967
COLOR_VER	0.6052998462939416	0.0009319765190218071	0.03052829047001825
ABS_UV	0.5111622465778148	0.0041343531654808915	0.06429893595916569
SDT	0.7780192543789404	0.00807688868347734	0.08987151207962031
SST	0.8052049368628316	0.0003694566851559964	0.019221256076437782
TURBIEDAD	0.8091109909453542	0.0004976272003723279	0.02230755926524298
TEMP_AMB	0.013382578721250529	0.0001875976485202529	0.11568309609122038

Tabla 5.9: Resultados obtenidos con ANN 1

Los resultados obtenidos con el segundo modelo de ANN solo muestran una mejora notable para el parámetro **N_TOTK** tal como se muestra en la **tabla 5.10**.

ANN 2	Métricas		
Y	R2	MSE	RMSE
COLI_FEC	-0,0012492445594667334	0.0009018217498842361	0.030030347148913148
COT	0.8933080505646943	0.00036253494335310296	0.01904035039995596
COT_SOL	0.8690962282009598	.0004379160122665108	0.020926442895688478
N_NH3	0.9969194934793607	1.9831941619416673e-05	0.0044533068184683475
N_NO2	-0,07328281378016843	0.0002602960059350573	0.016133691640013992
N_NO3	0.9959798029653354	1.281145761687859e-05	0.0035793096564670944
N_ORG	0.9742242878413037	3.54309729439469e-05	0.006137345088690312
N_TOT	0.9949077078123095	3.2980970505212665e-05	0.005742906102768237
N_TOTK	0.9985463604508859	9.709510219187702e-06	0.0031160087001142504
P_TOT	0.9364611774823347	0.0001796909737263115	0.01340488618848782
ORTO_PO4	0.8905222370186452	0.00020051351031493222	0.014160279316275235
COLOR_VER	0.619036957309083	0.0008995400864917049	0.029992333795350185
ABS_UV	0.5129886132383605	0.004118906640061168	0.06417870861945703
SDT	0.7822348266044424	0.007923502823333133	0.08901405969470853
SST	0.8123860717562381	0.00035583663621508577	0.01886363263571165
TURBIEDAD	0.8189350796893974	0.00047201685328048453	0.021725948846494245
TEMP_AMB	0.10093413107162785	0.013070211520887037	0.11432502578563906

Tabla 5.10: Resultados obtenidos con ANN 2

5.2.3. Resultados al agregar más parámetros

Posteriormente se agregaron 6 parámetros los cuales como se puede ver en la **gráfica 5.2** contienen de un 40 a 50 % de valores faltantes.

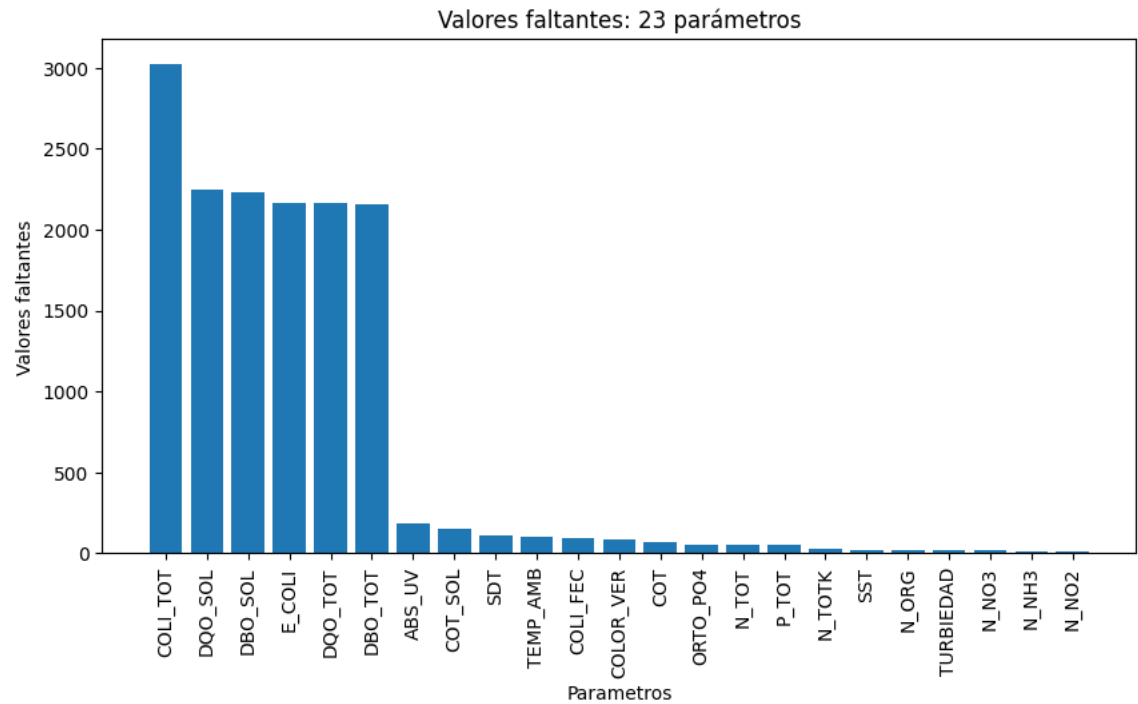


Figura 5.2: Valores faltantes con porcentajes de hasta 50 %

Los resultados obtenidos a través de **CNN** se muestran en la **tabla 5.11** y se lograron obtener **12** de **23** parámetros es decir alrededor de un **50 %** arriba de **0.8** para la métrica **R2**.

CNN	Métricas	
Y	R2	MSE
E_COLI	0.9652926520410156	6.398283464988238e-05
COLI_TOT	0.9612788875101885	9.677954863518686e-05
DBO_SOL	0.5808116216175357	0.0005769889319978316
DQO_SOL	0.6334474399889816	0.0009461396164336092
DQO_TOT	0.8052672561485998	0.00037329405689141804
DBO_TOT	0.8052672561485998	0.0002694258233642185
COLI_FEC	-0.007559942899116434	0.0009075057741674788
COT	0.8743923551471765	0.00042680971387674575
COT_SOL	0.8540695269886966	0.00048818525189160493
N_NH3	0.9995998771847171	2.5759440079486577e-06
N_NO2	-0.5392095407249384	0.00037329405689141804
N_NO3	0.9901924760288694	0.005590559728887884
N_ORG	0.9950689238164462	7.2059646237581405e-06
N_TOT	0.9218208107899076	0.0005063388820642076
N_TOTK	0.9988122966547189	7.93320309384626e-06
P_TOT	0.8888363934339495	0.0003143762492801728
ORTO_PO4	0.8810407220625038	0.0003364227985248641
COLOR_VER	0.5011689497594578	0.001177853166303535
ABS_UV	0.5155614856008873	0.004097146530654881
SDT	0.7706196114807364	0.008346128665616619
SST	0.6194312276238687	0.0007218030829506385
TURBIEDAD	0.7283041969155735	0.000708281857145111
TEMP_AMB	0.13029098340274148	0.012643434926628769

Tabla 5.11: Resultados finales obtenidos con CNN

Mientras que en la **tabla 5.12** se pueden observar los resultados obtenidos mediante ANN, donde se lograron obtener **15 de 23** parámetros es decir alrededor de un **65 %** arriba de **0.8** para la métrica **R2**, siendo estas las más efectivas para el tratamiento de datos faltantes en este conjunto de datos.

ANN	Métricas	
Y	R2	MSE
E_COLI	0.9652926520410156	6.398283464988238e-05
COLI_TOT	0.9605014031238532	9.872279311196233e-05
DBO_SOL	0.47824102233521737	0.0007181715209871788
DQO_SOL	0.6336707822833474	0.0009455631288687199
DQO_TOT	0.813336038856269	0.0002582621208375409
DBO_TOT	0.7549385871908376	0.000848611002295349
COLI_FEC	0.09257162971099175	0.0008173176112094741
COT	0.8790907591140487	3.54309729439469e-05
COT_SOL	0.8553812351239681	4.357912928947191e-06
N_NH3	0.997819350988004	1.5197839475785276e-05
N_NO2	-0.22712795381596051	0.0001531075265879536
N_NO3	0.9760356754008108	7.63688760208354e-05
N_ORG	0.9757544151315972	3.54309729439469e-05
N_TOT	0.9993271342345973	4.357912928947191e-06
N_TOTK	0.9977246864131795	1.5197839475785276e-05
P_TOT	0.945861098327577	0.0001531075265879536
ORTO_PO4	0.897574129203052	0.0001875976485202529
COLOR_VER	0.6052998462939416	0.0009319765190218071
ABS_UV	0.5111622465778148	0.0041343531654808915
SDT	0.7780192543789404	0.00807688868347734
SST	0.8052049368628316	0.0003694566851559964
TURBIEDAD	0.8091109909453542	0.0004976272003723279
TEMP_AMB	0.0794472034904844	0.013382578721250529

Tabla 5.12: Resultados finales obtenidos con ANN

Capítulo 6

CONCLUSIÓN

La reconstrucción de la información no es una tarea sencilla, se debe de conocer los datos en los cuales se trabaja, entender su comportamiento y así determinar porque faltan. No es posible realizar esta tarea y asegurar que se pueden obtener todos aquellos datos faltantes, no todas las variables serán candidatas algunas de ellas deberán ser descartadas en su totalidad. El objetivo de la imputación es estimar un posible valor para aquellos escenarios donde se tenga algún valor faltante y que sea posible utilizarlos para realizar un análisis como si no se tuviera la presencia de estos. Esta actividad no consiste en inventar datos sino que se determina en base a la información disponible que se tiene de cada variable, todo esto aplicando distintas estrategias. Este proceso va de la mano con la reconstrucción o predicción de estos valores, una red neuronal es capaz encontrar patrones y realiza un análisis más certero, más sin embargo utilizar modelos de aprendizaje máquina resulta una paradoja, puesto que el principal recurso para la generación de estos modelos son los datos, mismo del cual se carece y es aquí donde los datos imputados ayudan a solventar esta situación.

Emplear mal los métodos de imputación pueden conducir a sesgos y llevarnos a resultados negativos y pocos certeros, es por ello por lo que es importante analizar los patrones de los datos que se tienen para así seleccionar las técnicas más adecuadas.

6.1. Recomendaciones

Descartar aquellos parámetros que tengan valores muy repetitivos ya que solo sesgan los modelos de imputación, por ejemplo aquellas variables donde los datos tengan un comportamiento como el de la figura **4.11** del **capítulo 4**.

Realizar el proceso de imputación de manera cautelosa, revisar que la distribución de los datos imputados no haya sufrido alguna alteración después de dicha aplicación, puesto que esto conduce a resultados sesgados y por ende erróneos.

Comenzar la reconstrucción con aquellos variables que tenga un menor porcentaje de pérdida al 10 %. Al aplicar imputación múltiple lo más ideal es utilizar aquellas que tengan alguna correlación lineal. Por otro lado, la imputación simple puede ser útil para las variables que carecen de la misma.

Utilizar redes neuronales artificiales (ANN) ya que dan un mejor resultado, así mismo son útiles en variables que no tienen una correlación lineal.

Construir librerías para reutilizar código y agrupar las funciones necesarias para el proceso de imputación y de generación de modelos de aprendizaje máquina. Python es un lenguaje multiparadigma y brinda la posibilidad de adoptar las prácticas que el paradigma de la programación orientada a objetos ofrece, tal como lo es el caso expuesto.

REFERENCIAS

- Agatonovic-Kustrin, S. and Beresford, R. (2000). Basic concepts of artificial neural network (ann) modeling and its application in pharmaceutical research. *Journal of Pharmaceutical and Biomedical Analysis*, pages 717–727.
- Allison, P. D. (2009). Missing data. In *The SAGE handbook of quantitative methods in psychology*, pages 72–89. SAGE Publications.
- Amazon (2023). Amazon aws.
- Bar, H. (2017). Missing data — mechanisms and possible solutions / datos ausentes: mecanismos y posibles soluciones. *Culture and Education*, 29(3):492–525.
- Borana, J. (2016). Applications of artificial intelligence & associated technologies. In *Proceeding of International Conference on Emerging Technologies in Engineering, Biomedical, Management and*.
- CONAGUA (2023). Comisión nacional del agua | gobierno | gob.mx.
- de Madrid, U. C. I. (2023). Capítulo 17 análisis de correlación lineal los procedimientos correlaciones bivariadas y correlaciones parciales.
- Donders, T. A., van der Heijden, G. J., Stijnen, T., and Moons, K. G. (2006). Review: A gentle introduction to imputation of missing values. *Journal of Clinical Epidemiology*, pages 1087–1091.
- Dridi, S. (2021). Unsupervised learning - a systematic literature review.

-
- Gee Wah, N. and Wang Chi, L. (2020). Strong artificial intelligence and consciousness. *Journal of Artificial Intelligence and Consciousness*.
- Gholamy, A., Kreinovich, V., and Kosheleva, O. (2018). Why 70/30 or 80/20 relation between training and testing sets: A pedagogical explanation.
- Gogul, I. and Sathiesh Kumar, V. (2017). Flower species recognition system using convolution neural networks and transfer learning. In *2017 4th International Conference on Signal Processing, Communications and Networking*.
- Goldberg, S., Bolt, D., and Davidson, R. (2021). Data missing not at random in mobile health research: Assessment of the problem and a case for sensitivity analyses. *Journal of Medical Internet Research*.
- Harris, C. R., Millman, K. J., and van der Walt, S. J. (2020). Array programming with numpy. *Nature*, 585:357–362.
- Hartini, E. (2017). Classification of missing values handling method during data mining: Review.
- Heymans, M. and Twisk, J. (2022). Handling missing data in clinical research. *Journal of Clinical Epidemiology*, pages 185–188.
- Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9:90–95.
- Jierula, A., Wang, S., OH, T.-M., and Wang, P. (2021). Study on accuracy metrics for evaluating the predictions of damage locations in deep piles using artificial neural networks with acoustic emission data.
- Juna, A., Umer, M., Sadiq, S., and Karamti, H. (2022). Water quality prediction using knn imputer and multilayer perceptron. *Water*, 2022:2592.
- Kang, H. (2013). The prevention and handling of the missing data.

-
- Little, T., Jorgensen, T., Lang, K., and Moore, E. (2014). On the joys of missing data. *Journal of Pediatric Psychology*, 39(2):151–162.
- Mahesh, B. (2019). Machine learning algorithms - a review. 9(1):381–386.
- McKinney, W. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, pages 56–61.
- McKinney, W. (2013). *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. O'Reilly Media.
- Nakagawa, S. and Sosa, V. J. (2015). Missing data: Mechanisms, methods and messages. In *Ecological Statistics: Contemporary Theory and Application*. Oxford.
- Pandas (2023). pandas - python data analysis library.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., and Thirion, B. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.
- Python (2023). Python.
- Qiong, L. and Ying, W. (2012). Supervised learning. In *Encyclopedia of the Sciences of Learning*, pages 3243–3245.
- Ramesh, A., Kambhampati, C., Monson, J., and Drew, P. (2004). Artificial intelligence in medicine. *Ann R Coll Surg Engl*, pages 334–338.
- Russell, S. and Norvig, P. (2009). *Artificial Intelligence: A Modern Approach*. Pearson, third edition.
- Salgado, C. M., Azevedo, C., Proença, H., and Vieira, S. M. (2016). Missing data. In *Secondary Analysis of Electronic Health Records*, pages 143–162. Springer Open.

Sandoval, L. (2018). Machine learning algorithms for data analysis and prediction.

Sckit-learn (2023). scikit-learn: machine learning in python.

Senthilnathan, S. (2019). Usefulness of correlation analysis.

Wikipedia (2023a). Electrocardiografía.

Wikipedia (2023b). Tensorflow.

Wu, Y.-c. and Feng, J.-w. (2018). Development and application of artificial neural network. *Wireless Personal Communications*, pages 1645–1656.

Yamashita, R., Nishio, M., Gian Do, R. K., and Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, pages 611–629.

Zhang, S. (2012). Nearest neighbor selection for iteratively knn imputation. *The Journal of Systems and Software*, pages 2541–2552.

Zupan, J. (1994). Introduction to artificial neural network (ann) methods: what they are and how to use them. *Acta Chimica Slovenica*, pages 327–352.

Álvarez, I. (2022). *Introducción al Machine Learning*. Universidade de Santiago de Compostela, España.

ANEXOS A

Tabla de parámetros

CLAVE PARÁMETRO	PARÁMETRO	UNIDAD MEDIDA	CLAVE PARÁMETRO	PARÁMETRO	UNIDAD MEDIDA
ABS_UV	Absorción UV	U Abs/cm	CLORURO_DE_VINILO	Cloruro de Vinilo	mg/L
ACRILONITRILLO	Acrlonitrilo	mg/L	CLORUROS_TOT	Cloruros Totales	mg/L
ACROLEINA	Acroleína	mg/L	CN_TOT	Cianuros Totales	mg/L
ALC_FEN	Alcalinidad a la Fenoftaleína	mg CaCO3/L	CO3	Carbonatos	mg CaCO3/L
ALC_TOT	Alcalinidad Total	mg CaCO3/L	COLI_FEC	Coliformes Fecales	NMP/100 mL
ASTOT	Arsénico Total	mg/L	COLI_TOT	Coliformes Totales	NMP/100 mL
BENCENO	Benceno	mg/L	COLOR_VER	Color Verdadero	U Pt/Co
BR_CLMET	Bromoclorometano	mg/L	CONDUC_CAMPO	Conductividad eléctrica	µS/cm
BR_DICLMET	Bromodichlorometano	mg/L	CONDUC_CAMPO_FON	Conductividad eléctrica fondo	µS/cm
BROMOF	Bromoformo	mg/L	CONDUC_CAMPO_MED	Conductividad eléctrica media	µS/cm
BROMOMETANO	Bromometano	mg/L	CONDUC_CAMPO_SUP	Conductividad eléctrica superficie	µS/cm
CA_TOT	Calcio Total	mg/L	COT	Carbono Orgánico Total	mg/L
CAUDAL	Caudal	L/s	COT_SOL	Carbono Orgánico Soluble	mg/L
CD_TOT	Cadmio Total	mg/L	CR_TOT	Cromo Total	mg/L
CIS_1,2_DICLETEN	Cis-1,2-Dicloroetileno	mg/L	DBO_SOL	Demanda Bioquímica de Oxígeno soluble	mg/L
CL_BENC	Clorobenceno	mg/L	DBO_TOT	Demanda Bioquímica de Oxígeno Total	mg/L
CL_METILENO	Cloruro de Metileno	mg/L	DIBR_CLMET	Dibromoclorometano	mg/L
CLOROF_A	Clorofila A	mg/m3	DICLOROBENCENOS	Diclorobencenos (Mezcla de Isómeros)	mg/L
CLOROFORMO	Cloroformo	mg/L	DQO_SOL	Demanda Química de Oxígeno soluble	mg/L
CLOROMETANO	Clorometano	mg/L	DQO_TOT	Demanda Química de Oxígeno Total	mg/L
DUR_TOT	Dureza Total	mg CaCO3/L	N_ORG	Nitrógeno Orgánico	mg/L
E_COLI	Escherichia coli	NMP/100 mL	N_TOT	Nitrógeno Total (Cálculo)	mg/L
ENTEROC_FEC	Enterococos Fecales	NMP/100 mL	N_TOTK	Nitrógeno Kjeldahl	mg/L
ESTIRENO	Estireno	mg/L	NA_TOT	Sodio Total	mg/L
ETILBENCENO	Etilbenceno	mg/L	NL_TOT	Níquel Total	mg/L
FE_TOT	Hierro Total	mg/L	O_XILENO	o-Xileno	mg/L
FLUORUROS_TOT	Fluoruros Totales	mg/L	OD_%	Oxígeno Disuelto % Saturación	% Saturación
GRAS_ACEI	Grasas y Aceites	mg/L	OD_%_FON	Oxígeno Disuelto fondo % Saturación	% Saturación
H_HELMINTOS	Huevos de Helminto	No. de huevos / 5 L	OD_%_MED	Oxígeno Disuelto media % Saturación	% Saturación
HCO3	Bicarbonatos	mg CaCO3/L	OD_%_SUP	Oxígeno Disuelto superficie % Saturación	% Saturación
HG_TOT	Mercurio Total	mg/L	OD_mg/L	Oxígeno Disuelto	mg/L
K_TOT	Potasio Total	mg/L	OD_mg/L_FON	Oxígeno Disuelto fondo	mg/L
M_P_XILENO	m,p-Xilenos	mg/L	OD_mg/L_MED	Oxígeno Disuelto media	mg/L
MAT_FLOTANTE	MATERIA FLOTANTE	AUSENTE/PRESENTE	OD_mg/L_SUP	Oxígeno Disuelto superficie	mg/L
METILTERBUTILETER_MTBE	Metiliterbutiléter (MTBE)	mg/L	OH	Hidroxilos	mg CaCO3/L
MG_TOT	Magnesio Total	mg/L	ORTO_PO4	Fósforo Reactivo total (o-fosfatos)	mg/L
MN_TOT	Manganeso Total	mg/L	P_TOT	Fósforo Total	mg/L
N_NH3	Nitrógeno Amoniacal	mg/L	PB_TOT	Plomo Total	mg/L
N_NO2	Nitrógeno de Nitritos	mg/L	pH_CAMPO	Potencial de Hidrógeno	UpH
N_NO3	Nitrógeno de Nitratos	mg/L	pH_CAMPO_FON	Potencial de Hidrógeno fondo	UpH
pH_CAMPO_MED	Potencial de Hidrógeno media	UpH	TETRAACL_CARB	Tetracloruro de Carbono	mg/L
pH_CAMPO_SUP	Potencial de Hidrógeno superficie	UpH	TETRACLOROETILENO	Tetracloroetileno	mg/L

CLAVE PARÁMETRO	PARÁMETRO	UNIDAD MEDIDA	CLAVE PARÁMETRO	PARÁMETRO	UNIDAD MEDIDA
PO4_TOT	Fosfatos Totales	mg/L	THMs	Trihalometanos Totales	mg/L
POT_REDOX_CAMPO	Potencial REDOX	mV	TOLUENO	Tolueno	mg/L
POT_REDOX_CAMPO_FON	Potencial REDOX fondo	mV	TOX_D_48_FON_UT	Toxicidad Daphnia magna, Fondo, 48 h	UT
POT_REDOX_CAMPO_MED	Potencial REDOX media	mV	TOX_D_48_SUP_UT	Toxicidad Daphnia magna, Superficial, 48 h	UT
POT_REDOX_CAMPO_SUP	Potencial REDOX superficie	mV	TOX_D_48_UT	Toxicidad Daphnia magna, 48 h	UT
PROFUNDIDAD	Profundidad	m	TOX_FIS_FON_15_UT	Toxicidad Vibrio fischeri Fondo, 15min	UT
SAAM	Sustancias Activas al Azul de Metileno	mg/L	TOX_FIS_FON_30_UT	Toxicidad Vibrio fischeri Fondo, 30min	UT
SALINIDAD_FON	Salinidad fondo	‰	TOX_FIS_FON_5_UT	Toxicidad Vibrio fischeri Fondo, 5min	UT
SALINIDAD_MED	Salinidad media	‰	TOX_FIS_SUP_15_UT	Toxicidad Vibrio fischeri Superficial, 15min	UT
SALINIDAD_SUP	Salinidad superficie	‰	TOX_FIS_SUP_30_UT	Toxicidad Vibrio fischeri Superficial, 30min	UT
SDT	Sólidos Disueltos Totales	mg/L	TOX_FIS_SUP_5_UT	Toxicidad Vibrio fischeri Superficial, 5min	UT
SDT_M	Sólidos Disueltos Totales	mg/L	TOX_V_15_UT	Toxicidad Vibrio fischeri 15min	UT
SIO2	Silica Total (SiO2)	mg/L	TOX_V_30_UT	Toxicidad Vibrio fischeri, 30min	UT
SO4_TOT	Sulfatos Totales	mg/L	TOX_V_5_UT	Toxicidad Vibrio fischeri 5min	UT
SOLID_SED	Sólidos Sedimentables	mL/L	TRANS_1,2_DICLOROETILENO	Trans 1,2Dicloroetileno	mg/L
SST	Sólidos Suspendidos Totales	mg/L	TRANSPARENCIA	Transparencia disco de Secchi	m
TEMP_AGUA	Temperatura agua	°C	TRICLOROETILENO	Tricloroetileno	mg/L
TEMP_AMB	Temperatura ambiente	°C	TRICLOROFLUOROMETANO	Triclorofluorometano	mg/L
TRICLOROFLUOROMETANO	Triclorofluorometano	mg/L	TEMP_AGUA_3	Temperatura Prof. 3 m	°C
TURBIEDAD	Turbiedad	UNT	TEMP_AGUA_4	Temperatura Prof. 4 m	°C
XILEN_ISOMEROS	Mezcla de Isómeros de Xilenos	mg/L	TEMP_AGUA_5	Temperatura Prof. 5 m	°C
ZN_TOT	Zinc Total	mg/L	TEMP_AGUA_6	Temperatura Prof. 6 m	°C
1,1,1,2_TTCETA	1,1,1,2-Tetracloroetano	mg/L	TEMP_AGUA_7	Temperatura Prof. 7 m	°C
1,1,2,2_TETRACLETAN	1,1,2,2-Tetracloroetano	mg/L	TEMP_AGUA_8	Temperatura Prof. 8 m	°C
1,1,2_TCETA	1,1,2-Tricloroetano	mg/L	TEMP_AGUA_9	Temperatura Prof. 9 m	°C
1,1_DICLETILEN	1,1-Dicloroetileno	mg/L	TEMP_AGUA_10	Temperatura Prof. 10 m	°C
1,2,4_TRICLBENCEN	1,2,4-Triclorobenceno	mg/L	TEMP_AGUA_11	Temperatura Prof. 11 m	°C
1,2_DIBR_3_CLPROPAN	1,2-Dibromo-3-Cloropropano (DBCP)	mg/L	TEMP_AGUA_12	Temperatura Prof. 12 m	°C
1,2_DICLBENC	1,2-Diclorobenceno	mg/L	TEMP_AGUA_13	Temperatura Prof. 13 m	°C
1,2_DICLETAN	1,2-Dicloroetano	mg/L	TEMP_AGUA_14	Temperatura Prof. 14 m	°C
1,2_DICLPROPAN	1,2-Dicloropropano	mg/L	TEMP_AGUA_15	Temperatura Prof. 15 m	°C
1,3_DICLBEN	1,3-Diclorobenceno	mg/L	TEMP_AGUA_16	Temperatura Prof. 16 m	°C
1,4_DICLBEN	1,4-Diclorobenceno	mg/L	TEMP_AGUA_17	Temperatura Prof. 17 m	°C
2_CVEVE	2-Cloroetilvinil Éter	mg/L	TEMP_AGUA_18	Temperatura Prof. 18 m	°C
CU_TOT	Cobre Total	mg/L	TEMP_AGUA_19	Temperatura Prof. 19 m	°C
TEMP_AGUA_0.5	Temperatura Prof. 0.5 m	°C	TEMP_AGUA_20	Temperatura Prof. 20 m	°C
TEMP_AGUA_1	Temperatura Prof. 1 m	°C	TEMP_AGUA_21	Temperatura Prof. 21 m	°C
TEMP_AGUA_2	Temperatura Prof. 2 m	°C	TEMP_AGUA_22	Temperatura Prof. 22 m	°C

CLAVE PARÁMETRO	PARÁMETRO	UNIDAD MEDIDA	CLAVE PARÁMETRO	PARÁMETRO	UNIDAD MEDIDA
TEMP_AGUA_23	Temperatura Prof. 23 m	°C	TEMP_AGUA_SUP	Temperatura agua superficie	°C
TEMP_AGUA_24	Temperatura Prof. 24 m	°C	1,2_DIFENILHIDRACINA	1,2-Difenilhidracina	mg/L
TEMP_AGUA_25	Temperatura Prof. 25 m	°C	1_C_NAF	1-Cloronaftaleno	mg/L
TEMP_AGUA_26	Temperatura Prof. 26 m	°C	2,3_DICLOROFENOL_DCF	2,3-Diclorofenol	mg/L
TEMP_AGUA_27	Temperatura Prof. 27 m	°C	2,4,5_TRICLOROFENOL	2,4,5-Triclorofenol	mg/L
TEMP_AGUA_28	Temperatura Prof. 28 m	°C	2,4,6_TRICLFENOL	2,4,6-Triclorofenol	mg/L
TEMP_AGUA_29	Temperatura Prof. 29 m	°C	2,4_DICLFENOL	2,4-Diclorofenol	mg/L
TEMP_AGUA_30	Temperatura Prof. 30 m	°C	2,4_DIMETILFEN	2,4-Dimetilfenol	mg/L
TEMP_AGUA_31	Temperatura Prof. 31 m	°C	2,4_DINITOL	2,4-Dinitrotolueno	mg/L
TEMP_AGUA_32	Temperatura Prof. 32 m	°C	2,4_DNF	2,4-Dinitrofenol	mg/L
TEMP_AGUA_33	Temperatura Prof. 33 m	°C	2,6_DINTOL	2,6-Dinitrotolueno	mg/L
TEMP_AGUA_34	Temperatura Prof. 34 m	°C	2_CF	2-Clorofenol	mg/L
TEMP_AGUA_35	Temperatura Prof. 35 m	°C	2_CLNAFT	2-Cloronaftaleno	mg/L
TEMP_AGUA_36	Temperatura Prof. 36 m	°C	2_NIFENOL	2-Nitrofenol	mg/L
TEMP_AGUA_37	Temperatura Prof. 37 m	°C	4_BRFENFENETER	4-Bromofenil Fenil Éter	mg/L
TEMP_AGUA_38	Temperatura Prof. 38 m	°C	4_NITROFENOL	4-Nitrofenol	mg/L
TEMP_AGUA_39	Temperatura Prof. 39 m	°C	ACENAF	Acenafteno	mg/L
TEMP_AGUA_39.5	Temperatura Prof. 39.5 m	°C	ACENAFTIL	Acenaftileno	mg/L
TEMP_AGUA_FON	Temperatura agua fondo	°C	AL_TOT	Aluminio Total	mg/L
TEMP_AGUA_MED	Temperatura agua media	°C	ANTRA	Antraceno	mg/L
B2_ETILHEXIL_FTALATO	Bis (2-Etilhexil) Ftalato	mg/L	HEXACL_BUT	Hexaclorobutadieno	mg/L
BEN_A_ANT	Benzo(A)Antraceno	mg/L	HEXACL_CICLOPEN	Hexaclorociclopentadieno	mg/L
BEN_A_PIR	Benzo(A)Pireno	mg/L	HEXACL_ETANO	Hexacloroetano	mg/L
BEN_B_FANT	Benzo(B)Fluoranteno	mg/L	IND_PIR	Indeno (1,2,3-CD) Pireno	mg/L
BEN_G_HI_PER	Benzo(GHI)Perileno	mg/L	ISO_FOR	Isoforona	mg/L
BEN_K_FANT	Benzo(K)Fluoranteno	mg/L	N_NITROS_DIMETILA	N-Nitrosodimetilamina	mg/L
BENCIDINA	Bencidina	mg/L	N_NITROS_N_PROPAM	N-Nitrosodi-N-Propilamina	mg/L
BIS2_CLETETER	Bis (2-Cloroetil) Éter	mg/L	N_NITROSODIFENILAMINA	N-Nitrosodifenilamina	mg/L
BIS2_CLIOISOPET	Bis (2-Cloroisopropil) Éter	mg/L	NAFTALENO	Naftaleno	mg/L
CRISENO	Criseno	mg/L	NITROBENCENO	Nitrobenzeno	mg/L
DI_2_ETILEXIL_ADIPATO	Di (2 Etilxil) Adipato	mg/L	PCBEN	Pentaclorobenzeno	mg/L
DIBEN_ANT	Dibenzo(AH)Antraceno	mg/L	PENTACLOROFENOL	Pentaclorofenol	mg/L
DIET_FTAL	Diethylftalato	mg/L	PIRENO	Pireno	mg/L
DIMETIL_FTAL	Dimetilftalato	mg/L	SE_TOT	Selenio Total	mg/L
DINITRO_O_CRESOL	Dinitro-o-Cresol	mg/L	SULFUROS_TOT	Sulfuros (H2S)	mg/L
FENANTRENO	Fenantreno	mg/L	BA_TOT	Bario Total	mg/L
FENOL	Fenol	mg/L	HAPS	Total de hidrocarburos poliaromáticos	mg/L
FENOLES_TOT	Fenoles Totales	mg/L	TALIO	Talio Total	mg/L
FLUORANTENO	Fluoranteno	mg/L	CLORUROS_SOL	Cloruros Solubles	mg/L
FLUORENO	Fluoreno	mg/L	FLUORUROS_SOL	Fluoruros Solubles	mg/L

ANEXOS B

Matriz de correlación entre parámetros

COLI_FEC	1	0.16	0.17	0.27	0.002	-0.015	0.2	0.28	0.28	0.26	0.24	0.082	0.11	-0.043	0.035	0.017	-0.018
COT	0.16	1	0.95	0.57	0.056	-0.038	0.42	0.58	0.6	0.56	0.53	0.26	0.5	-0.023	0.092	0.11	0.032
COT_SOL	0.17	0.95	1	0.56	0.048	-0.04	0.4	0.57	0.59	0.55	0.52	0.28	0.5	-0.05	0.092	0.11	0.028
N_NH3	0.27	0.57	0.56	1	0.12	-0.034	0.42	0.94	0.96	0.89	0.85	0.22	0.37	-0.13	0.051	0.076	0.062
N_NO2	0.002	0.056	0.048	0.12	1	0.22	0.23	0.24	0.18	0.32	0.2	0.043	0.074	-0.085	0.034	0.056	-0.0037
N_NO3	-0.015	-0.038	-0.04	-0.034	0.22	1	-0.047	0.2	-0.043	0.048	0.066	0.03	0.0016	-0.18	0.037	0.063	-0.017
N_ORG	0.2	0.42	0.4	0.42	0.23	-0.047	1	0.63	0.66	0.61	0.45	0.24	0.35	0.14	0.22	0.21	0.029
N_TOT	0.28	0.58	0.57	0.94	0.24	0.2	0.63	1	0.97	0.92	0.84	0.26	0.41	-0.1	0.12	0.14	0.057
N_TOTK	0.28	0.6	0.59	0.96	0.18	-0.043	0.66	0.97	1	0.93	0.84	0.26	0.41	-0.061	0.11	0.13	0.062
P_TOT	0.26	0.56	0.55	0.89	0.32	0.048	0.61	0.92	0.93	1	0.88	0.31	0.38	-0.15	0.21	0.22	0.072
ORTO_PO4	0.24	0.53	0.52	0.85	0.2	0.066	0.45	0.84	0.84	0.88	1	0.21	0.35	-0.14	0.05	0.084	0.074
COLOR_VER	0.082	0.26	0.28	0.22	0.043	0.03	0.24	0.26	0.26	0.31	0.21	1	0.34	-0.15	0.35	0.56	0.089
ABS_UV	0.11	0.5	0.5	0.37	0.074	0.0016	0.35	0.41	0.41	0.38	0.35	0.34	1	0.017	0.14	0.2	0.07
SDT	-0.043	-0.023	-0.05	-0.13	-0.085	-0.18	0.14	-0.1	-0.061	-0.15	-0.14	-0.15	0.017	1	-0.035	-0.11	-0.077
SST	-0.035	0.092	0.092	0.051	0.034	0.037	0.22	0.12	0.11	0.21	0.05	0.35	0.14	-0.035	1	0.86	0.011
TURBIEDAD	0.017	0.11	0.11	0.076	0.056	0.063	0.21	0.14	0.13	0.22	0.084	0.56	0.2	-0.11	0.86	1	0.058
TEMP_AMB	-0.018	0.032	0.028	0.062	-0.0037	-0.017	0.029	0.057	0.062	0.072	0.074	0.089	0.07	-0.077	0.011	0.058	1
COLI_FEC		COT	COT_SOL	N_NH3	N_NO2	N_NO3	N_ORG	N_TOT	N_TOTK	P_TOT	ORTO_PO4	COLOR_VER	ABS_UV	SDT	SST	TURBIEDAD	TEMP_AMB