Universitat Oberta de Catalunya

uoc.edu

Técnicas de aprendizaje automático para el análisis de la salinidad de aguas en el valle del Guadalhorce

Carlos A. Alonso Cabrera
Master en Ciencia de Datos
Área 2

Carlos Luis Sánchez Bocanegra
Rafael Pastor Vargas

Jordi Casas Roma

Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada 3.0 España de Creative Commons

Copyleft (ɔ) 2020 Carlos Alberto Alonso Cabrera

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Técnicas de aprendizaje automático para el análisis de la salinidad de aguas en el valle del Guadalhorce*
Nombre del autor:	*Carlos Alberto Alonso Cabrera*
Nombre del consultor/a:	*Carlos Luis Sánchez Bocanegra* *Rafael Pastor Vargas*
Nombre del director:	*Jordi Casas Roma*
Fecha de entrega (mm/aaaa):	27/09/2020
Titulación:	*Master en Ciencia de Datos*
Área del Trabajo Final:	*Área 2*
Idioma del trabajo:	*Castellano*
Palabras clave	*Salinidad, análisis de datos, presa*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

La Junta de Andalucía es la encargada de administrar y operar tres presas (presa del Conde de Guadalhorce, Guadalhorce y Guadalteba) que actúan sobre el río Guadalhorce, y sus afluentes Gudadalteba y Turón.

Existen datos recogidos sobre dichas presas desde 1974. La necesidad principal del trabajo surge de la inquietud de la Junta de Andalucía por conocer lo que muestran los datos recogidos, y en particular, los datos sobre la salinidad de la presa Guadalhorce y como influye en el resto de presas.

En la actualidad de se mezcla el agua de las tres presas, dependiendo de su capacidad, para obtener agua para consumo, regadío, ganadería, etc.

Lo que se pretende es aplicar métodos de aprendizaje automático para conocer que parámetros influyen en la salinidad de la mezcla y cual es la mezcla óptima de las tres presas de cara a sesgar agua para consumo, regadío y ganadería.

Utilizar la menor cantidad de agua dulce, proveniente de Guadalteba y Conde de Guadalhorce, para que la mezcla fuera utilizable.

Como influyen parámetros, como la altura, lluvia, etc, en la salinidad del agua de la presa Guadalhorce.

Relacionar la salinidad de la presa Guadalhorce la salinidad que se obtiene de la mezcla.

Se tomarán los datos proporcionados por la Junta de Andalucía, que recogen diferente información acerca de las presas, se preprocesarán, se analizarán y se visualizarán para extraer un análisis pormenirizado.

Abstract (in English, 250 words or less):

The Junta de Andalucía is in charge of managing and operating three dams (Conde de Guadalhorce, Guadalhorce and Guadalteba Dam) that act on the Guadalhorce River, and its Guadalteba and Turón tributaries.

There are data collected on these dams since 1974. The main need for the work arises from the concern of the Junta de Andalucía to know what the data collected shows, and in particular, the data on the salinity of Guadalhorce dam and how it influences in the rest of the dams.

At present, the water from the three dams is mixed, according to their capacity, to obtain water for consumption, irrigation, livestock, etc.

The aim is to apply Machine Learning methods to know what parameters influence the mixture salinity and what is the optimal mixture of the three dams in order to skew water for consumption, irrigation and livestock.

Use the least amount of fresh water, province of Guadalteba and Conde de Guadalhorce, so that the mixture is usable.

How parameters, such as height, rain, etc., influence the salinity of the water in the Guadalhorce dam.

Relate the salinity of the Guadalhorce dam to the salinity obtained from the mixture.

The data provided by the Junta de Andalucía, which collect different information on the dams, will be pre-processed, analyzed and displayed to extract a detailed analysis.

Agradecimientos

A mi familia, por su apoyo incondicional.

A Dolores Fernández Carmona, por su continua colaboración y compromiso. Sin ella no hubiera sido posible realizar este estudio.

A Rafal Pastor Vargas, por compartir experiencia y conocimientos cuando mas lo necesitaba.

Y por último, a mi tutor, Carlos Sánchez Bocanegra, por su confianza y paciencia durante todo el proceso.

Índice

1. Introducción 1

1.1 Contexto y justificación del Trabajo 1

1.2 Objetivos del Trabajo 1

1.3 Enfoque y método seguido 2

1.4 Planificación del Trabajo 2

1.5 Breve sumario de productos obtenidos 2

1.6 Breve descripción de los otros capítulos de la memoria 2

2. Análisis del ámbito y estado del arte 3

2.1 ¿Que es salinidad? 3

2.1.1 Salinidad primaria 3

2.1.2 Salinidad secundaria 4

2.2 El impacto de la salinidad 5

2.2.1 Efecto de la salinidad en la fisiología de humanos e insectos 6

2.2.2 Efecto de la salinidad en las plantas 7

2.3 Problemática de la salinidad en el valle del Guadalhorce y solución actual 8

2.4 Aplicación de métodos de aprendizaje automático a la calidad del agua 11

2.5 Preguntas a responder 13

3. Proceso de implementación 13

3.1 Captura de datos 14

3.2 Almacenamiento y descripción del conjunto de datos 16

3.3 Limpieza, preprocesado y preparación de los datos 24

3.3.1 Integración de los datos 25

3.3.2 Limpieza de los datos 26

3.3.2.1 Gestión de *missing values* en variable etiquetada 27

3.3.3 Transformación de los datos 37

3.3.3.1 Discretización de la variable etiquetada 37

3.3.3.2 Normalización del conjunto de datos 38

3.3.4 Reducción de datos 39

3.4 Métodos de aprendizaje automático 40
3.4.1 Árboles de decisión 43
3.4.2 Random Forest 44
3.4.3 Support Vector Machines 45
3.4.4 AdaBoost 46
3.4.5 Redes Neuronales Artificiales (ANN) 47
3.5 Análisis de los resultados 50
4. Conclusiones y líneas de futuro 59
4.1 Conclusiones 59
4.2 Líneas futuras 61
5. Glosario 63
6. Bibliografía 65
7. Anexos 71
7.1 Anexo 1. Código Python 71

Lista de figuras

Figura 1: Planificación propuesta 2

Figura 2: Hidrografía de la región del Trías de Antequera 8

Figura 3: Valle del Guadalhorce y sistema de presas 10

Figura 4: Pirámide DIKW 14

Figura 5: Estadillo mensual Agencia Andaluza del Agua 2012 16

Figura 6: Esquema de 5 estrellas para la publicación de los datos 16

Figura 7: Dataframe sistema de presas valle del guadalhorce consolidado 26

Figura 8: Missing values presentes en el dataset 28

Figura 9: Porcentaje de missing values en Salinidad_mezcla 29

Figura 10: Imputación missing values en variable Salinidad_mezcla con función mean() 29

Figura 11: Distribución de Salinidad_mezcla con imputación con función mean() 30

Figura 12: Imputación missing values en variable Salinidad_mezcla con función median() 30

Figura 13: Distribución de Salinidad_mezcla con imputación con función median() .. 30

Figura 14: Imputación missing values en variable Salinidad_mezcla con función rolling mean() 31

Figura 15: Distribución de Salinidad_mezcla con imputación con función rolling mean() 31

Figura 16: Imputación missing values en variable Salinidad_mezcla con función rolling median() 32

Figura 17: Distribución de Salinidad_mezcla con imputación con función rolling median() 32

Figura 18: Imputación missing values en variable Salinidad_mezcla con interpolación lineal 33

Figura 19: Distribución de Salinidad_mezcla con imputación con interpolación lineal 33

Figura 20: Imputación missing values en variable Salinidad_mezcla con interpolación cuadrática 33

Figura 21: Distribución de Salinidad_mezcla con imputación con interpolación cuadrática 34

Figura 22: Imputación missing values en variable Salinidad_mezcla con interpolación cúbica 34

Figura 23: Distribución de Salinidad_mezcla con imputación con interpolación cúbica 34

Figura 24: Imputación missing values en variable Salinidad_mezcla con interpolación spline 35

Figura 25: Distribución de Salinidad_mezcla con imputación con interpolación spiline 35

Figura 26: Imputación missing values en variable Salinidad_mezcla con interpolación Akima 36

Figura 27: Distribución de Salinidad_mezcla con imputación con interpolación Akima 36

Figura 28: Predicción de missing values en variable Salinidad_mezcla mediante Regresion Lineal 37

Figura 29: Distribución de Salinidad_mezcla con Regresion Lineal 37

Figura 30: Univariate Feature Selection con SelectKBest() y test ANOVA F-value 40

Figura 31: Division del dataset 41

Figura 32: Validación cruzada de 5 pliegues o 5-folds cross-validation 42

Figura 33: Estructura de los árboles de decisión 43

Figura 34: Vectores de soporte y margen en SVM 45

Figura 34: Red multicapa feed-forward 48

Figura 35: Deep learning vs métodos tradicionales [60] 51

Figura 36: Matriz de confusión del clasificador Random Forest 52

Figura 37: Valores de Salinidad_mezcla para el año 2012 52

Figura 38: Valores de Salinidad_mezcla con interpolación lineal para el año 2012 ... 53

Figura 39: Tunnig de Hiperparámetros para RandomForestClassifier() 54

Figura 40: Fronteras de decisión de RandomForestClassifier 55

Figura 41: Predicción de Salinidad_mezcla en función de Salinidad_guadalhorce 56

Figura 42: Predicción de Salinidad_mezcla en función de Cota_guadalhorce 56

Figura 43: Predicción de Salinidad_mezcla en función de la suma de los volúmenes embalsados de Guadalhorce, Guadalteba y Conde del Guadalhorce 57

Figura 44: Predicción de Salinidad_mezcla en función del desembalse de Guadalteba 57

Figura 45: Predicción de Salinidad_mezcla en función del desembalse de Guadalhorce 58

Figura 46: Areas de Data Governance según DAMA 62

Lista de tablas

Tabla 1: Estimación mundial de la salinización secundaria en las tierras de regadío del mundo [5] *5*

Tabla 2: Precisión de clasificación en los conjuntos de entrenamiento y validación *50*

Tabla 3: Precisión de clasificación en los conjuntos de test *51*

Lista de ecuaciones

Ecuación 1: Expresión matemática de normalización min-max *38*

1. Introducción

1.1 Contexto y justificación del Trabajo

La Junta de Andalucía es la encargada de administrar y operar tres presas (presa del Conde de Guadalhorce, Guadalhorce y Guadalteba) que actúan sobre el río Guadalhorce, y sus afluentes Guadalteba y Turón.

En la actualidad se mezcla el agua de las tres presas, dependiendo de su capacidad, para obtener agua para consumo, regadío, ganadería, etc. Esta mezcla se realiza de forma intuitiva, recurriendo a formulas de proporción.

Existen datos recogidos sobre dichas presas desde 1974. La necesidad principal del trabajo surge de la inquietud de la Junta de Andalucía por conocer lo que muestran los datos recogidos, y en particular, los datos sobre la salinidad de una de la presa Guadalhorce y como influye en el resto de presas.

Lo que se pretende es aplicar métodos de aprendizaje automático (Machine Learning) para conocer que parámetros influyen en la salinidad de la mezcla y cual es la mezcla óptima de las tres presas de cara a sesgar agua para consumo, regadío y ganadería.

1.2 Objetivos del Trabajo

Objetivo principal (OP): ¿Es posible hallar una mezcla óptima de agua usable maximizando la cantidad de agua de la presa Guadalhorce? Si es posible, ¿Qué parámetros son necesarios para lograr esta mezcla óptima?

Objetivo secundario 1 (OS1): Utilizar la menor cantidad de agua dulce, proviniente de Guadalteba y Conde de Guadalhorce, para que la mezcla fuera utilizable.

Objetivo secundario 2 (OS2): Estudiar parámetros, como la altura, lluvia, etc, en la salinidad del agua de la presa Guadalhorce.

Objetivo secundario 3 (OS3): Relacionar la salinidad de la presa Guadalhorce la salinidad que se obtiene de la mezcla.

1.3 Enfoque y método seguido

Se recopilarán los datos proporcionados por la Junta de Andalucía, que recogen diferente información acerca de las presas, se preprocesarán, se analizarán y se visualizarán para extraer un análisis pormenirizado.

1.4 Planificación del Trabajo

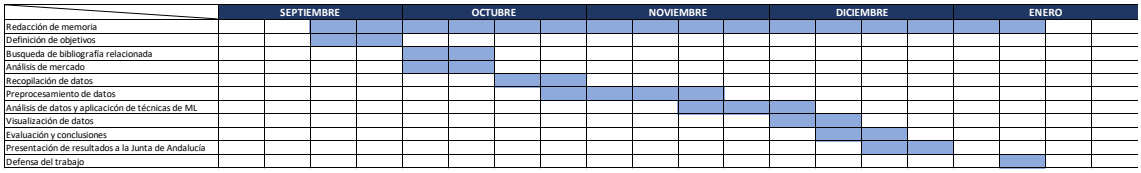


Figura 1: Planificación propuesta

1.5 Breve sumario de productos obtenidos

El resultado una serie de algoritmos cuyo objetivo principal será cargar, preprocesar, analizar y visualizar los datos proporcionados por la Junta de Andalucía de cara a sacar conclusiones sobre la salinidad de la presa de Guadalhorce.

Adicionalmente se analizarán los posibles errores en la toma de esos datos.

1.6 Breve descripción de los otros capítulos de la memoria

En los siguientes capítulos se contemplarán; funcionamiento del sistema actual, estudio del arte, descripción de los datos actuales, preprocesamiento de los datos, análisis de los datos, visualización de los datos, evaluación, conclusiones y mejoras futuras.

2. Análisis del ámbito y estado del arte

2.1 ¿Que es salinidad?

La salinidad es una medida del contenido de sales en el suelo o el agua. Las sales son muy solubles en aguas superficiales y subterráneas, y pueden transportarse a través del agua [1].

La sal en nuestros recursos hídricos generalmente se deriva de tres fuentes. En primer lugar, pequeñas cantidades de sal (principalmente cloruro de sodio) se evaporan del agua del océano y se transportan en nubes de lluvia y se depositan a través del paisaje con la lluvia. En segundo lugar, algunos paisajes también pueden contener sal que se ha desprendido de las rocas durante los procesos meteorológicos (degradación gradual) y, en tercer lugar, la sal puede permanecer en los sedimentos que dejaron los mares tras su retirada después de períodos en los que los niveles del océano eran mucho más altos o la superficie terrestre mucho más baja [2].

La salinidad puede tomar dos formas, clasificadas según sus causas; salinidad primaria y salinidad secundaria.

2.1.1 Salinidad primaria

La salinidad primaria resulta de la acumulación de sales durante largos períodos de tiempo, a través de procesos naturales, en el suelo o en las aguas subterráneas. Es causada por dos procesos naturales.

En primer lugar, el desgaste de los materiales parentales que contienen sales solubles. Los procesos meteorológicos descomponen las rocas y liberan sales solubles de varios tipos, principalmente cloruros de sodio, calcio y magnesio y, en menor medida, sulfatos y carbonatos. El cloruro de sodio es la sal más soluble [3].

En segundo lugar, la deposición de sal oceánica transportada por el viento y la lluvia. Las sales cíclicas son sales oceánicas transportadas tierras adentro por el viento y depositadas por la lluvia, principalmente cloruro de sodio (NaCl), aunque también están presentes iones de sulfato (SO_4), magnesio (Mg), calcio (Ca) o potasio (K). La cantidad de sales almacenada en el suelo varía con el tipo de suelo, siendo baja para suelos arenosos y alta para suelos que contienen un alto porcentaje de minerales arcillosos. También varía inversamente con la precipitación media anual [3].

2.1.2 Salinidad secundaria

La salinización secundaria es el resultado de actividades humanas que cambian el equilibrio hidrológico del suelo. Las causas más comunes son; la eliminación o el reemplazo de vegetación autóctona por cultivos estacionales, y el uso de esquemas de riego que utilizan agua de riego rica en sal o que tienen un drenaje insuficiente [3, 4].

Antes de las actividades humanas, en climas áridos o semiáridos, el agua utilizada por la vegetación natural estaba en equilibrio con la lluvia, con las raíces profundas de la vegetación nativa asegurando que los niveles freáticos estuvieran muy por debajo de la superficie. En su estado natural, la vegetación nativa de raíces profundas y perenne utiliza casi toda el agua de lluvia que cae sobre la tierra. En climas áridos o semiáridos, la tasa de crecimiento de la vegetación natural está limitada por la disponibilidad de agua de lluvia. Las sales serán arrastradas por la lluvia y se acumularán en el fondo de la zona de las raíces hasta la concentración límite para que las raíces extraigan agua, aproximadamente a 50 dS / m [3, 4].

La eliminación de vegetación y el riego cambiaron este equilibrio, de modo que la lluvia por un lado y el agua de riego por el otro proporcionaron más agua de la que los cultivos podían utilizar. El exceso de agua eleva el nivel freático, moviliza las sales previamente almacenadas en el subsuelo y las lleva hasta la zona superficial. Las plantas usan el agua y dejan la sal hasta que el agua del suelo se vuelve demasiado salina para que las raíces la absorban. El nivel freático sigue aumentando y, cuando se acerca a la superficie, el agua se evapora dejando sales en la superficie y, por lo tanto, formando acumulaciones de sal en el suelo. La sal movilizada también puede desplazarse lateralmente a los cursos de agua y aumentar su salinidad [4].

En 1987, la tierra dedicada a regadío sumaba un total de 227 millones de hectáreas en todo el mundo [5] (tabla 1). En muchas áreas irrigadas, el nivel freático ha aumentado debido a las cantidades excesivas de agua aplicada junto con un drenaje deficiente. En la mayoría de los proyectos de riego ubicados en áreas semiáridas y áridas, los problemas de anegamiento y salinidad del suelo han alcanzado proporciones graves incluso antes de que se pudiera realizar todo el potencial del proyecto de riego. La mayoría de los sistemas de riego del mundo han provocado salinidad secundaria, sodicidad o anegamiento. La tabla 1 muestra que la proporción de tierras de regadío afectadas por la sal en varios países oscila entre un mínimo del 9% y un máximo del 34%, con un promedio mundial del 20%. La tierra de regadío representa solo el 15% del total de la tierra cultivada, pero como la tierra de regadío tiene al menos el doble de productividad que la de secano, puede producir un tercio de los alimentos del mundo [4, 5].

Country	Total de tierra cultivada Mha	Area dedicada a riego		Area de tierra dedicadas a riego afectadas por salinidad	
		Mha	%	Mha	%
China	97	45	46	6.7	15
India	169	42	25	7.0	17
Soviet Union	233	21	9	3.7	18
United States	190	18	10	4.2	23
Pakistan	21	16	78	4.2	26
Iran	15	6	39	1.7	30
Thailand	20	4	20	0.4	10
Egypt	3	3	100	0.9	33
Australia	47	2	4	0.2	9
Argentina	36	2	5	0.6	34
South Africa	13	1	9	0.1	9
Subtotal	843	159	19	29.6	20
World	1,474	227	15	45.4	20

Tabla 1: Estimación mundial de la salinización secundaria en las tierras de regadío del mundo [5]

El agua de riego agrega cantidades apreciables de sal, incluso con agua de riego de buena calidad que contiene solo 200-500 mg / kg de sal soluble. El agua de riego con un contenido de sal de 500 mg / kg (es decir, 500 mg / L) contiene 0,5 toneladas de sal por 1000 m3. Dado que los cultivos requieren de 6.000 a 10.000 m3 de agua por hectárea cada año, una hectárea de tierra recibirá de 3 a 5 toneladas de sal. Debido a que la cantidad de sal eliminada por los cultivos es insignificante, la sal se acumulará en la zona de las raíces y debe disolverse proporcionando más agua de la que requieren los cultivos. Si el drenaje no es el adecuado, el exceso de agua hace que suba el nivel freático, movilizando las sales que se acumulan en la zona superficial. Cuando el cultivo no puede utilizar toda el agua aplicada, se produce un anegamiento [6, 7].

2.2 El impacto de la salinidad

El impacto de una alta salinidad en el agua proveniente de reservas es múltiple; producción agrícola, calidad del agua, salud ecológica de los arroyos, biodiversidad, erosión del suelo, riesgo de inundación, deterioro en las infraestructuras de las reservas, irrigación, etc.

En este capitulo nos centraremos en el impacto de la salinidad en la fisiología de humanos y algunos insectos, y en el efecto sobre las platas/cultivos.

2.2.1 Efecto de la salinidad en la fisiología de humanos e insectos

Los efectos sobre la salud de la población a largo plazo del consumo de cantidades sustanciales de sodio a través del agua potable siguen siendo desconocidos [8] aunque hay distintos estudios que señalan su efecto perjudicial.

Un estudio publicado en Nature (2019) mostró que el consumo de Na+ tiene efectos inmunológicos en el tejido de la piel, microbiología intestinal, y otros órganos, así como enfermedades cardiovasculares, inflamación, infección y autoinmunidad [9].

Otro estudio señala que alrededor del 20 por ciento de los adultos y del 40 al 65 por ciento de los ancianos personas en Bangladesh sufren de hipertensión, que es un problema médico y público cada vez más importante problema de salud [10]. Se ha demostrado que el aumento de la ingestión de sodio en la dieta contribuye al riesgo de hipertensión [8].

Debido al uso de agua con alto contenido en sodio, los ciudadanos de en la costa sur de Bangladesh padecen numerosas enfermedades que incluían afecciones de la piel, caída del cabello, diarrea, enfermedades gástricas e hipertensión arterial [11].

Un estudio que relaciona la salinidad del agua de consumo y su riesgo para las mujeres embarazadas, señala que el sodio del agua potable tiene serias implicaciones para la salud de la comunidad, particularmente para las mujeres embarazadas. Las personas expuestas a concentraciones de agua potable ligeramente salinas (rango entre 1000-2000 mg / L) y moderadamente salinas (2000 mg / L) tenían, respectivamente, un 17% y un 42% de probabilidades más altas de ser hipertensos que los que consumían agua dulce (<1000 mg / L) [8]. Se encuentra una relación sustancial entre beber agua con alto nivel de sodio y la pre-eclampsia (complicación del embarazo caracterizada por presión arterial alta y signos de daños en otro sistema de órganos [12]) e hipertensión gestacional [13].

Pero, ¿Cuáles son las cantidades recomendadas de sales recomendadas por las organizaciones internacionales? De acuerdo con la Organización Mundial de la Salud (OMS) y la consulta conjunta de expertos de la Organización de las Naciones Unidas para la Agricultura y la Alimentación (FAO) (2002), la ingesta nutricional de sodio es de 2 g / día [14]. Basado en el gusto o sabor, y relativo a la medida de total de sales disueltas (TDS por sus siglas en inglés) por litro, un agua de menos de 600 mg / L se considera agua potable de buena calidad, un agua que contenga entre 600 y 900 mg / L se considera de calidad regular, un agua que contenga entre 900 y 1200 mg / L se considera de mala calidad y, por último, un agua que contenga más de 1200 mg / L se considera inaceptable [15].

La osmorregulación animal (proceso que mantiene el balance de agua y sales (balance osmótico) a través de todas las membranas del cuerpo [16]) se ha estudiado ampliamente. En uno de estos estudios se sostiene que las especies de efímeras pueden sufrir una mortalidad sustancial a una osmolalidad (es decir, concentración osmótica) inferior a la de su fluido interno. Esto podría estar relacionado con el aumento de la captación de iones, la pérdida de la regulación del pH o el envenenamiento por Na [17].

En el mismo sentido, se observa que niveles elevados de sulfatos, en especies de efímeras *Neocloeon triangulifer*, impone una demanda energética asociada con el mantenimiento de la homeostasis (propiedad de los organismos que consiste en su capacidad de mantener una condición interna estable) que se manifiesta principalmente en tasas de crecimiento reducidas y asociadas retrasos del desarrollo. También identifican dos genes relacionados con transporte de sulfato en esta especie, que puede ser una herramienta prometedora para investigar los mecanismos de toxicidad de los sulfatos [18].

2.2.2 Efecto de la salinidad en las plantas

La salinidad del suelo es un factor importante que limita el rendimiento de los cultivos agrícolas, poniendo en peligro la capacidad de la agricultura para sostener el creciente aumento de la población humana [19, 20, 21]. A bajas concentraciones de sal, los rendimientos se ven levemente afectados o no se ven afectados en absoluto [22]. A medida que aumentan las concentraciones, los rendimientos se acercan a cero, ya que la mayoría de las plantas, incluidas la mayoría de las plantas de cultivo, no crecerán en altas concentraciones de sal y se inhibirán gravemente o incluso se eliminarán con NaCl 100-200 mM. La razón es que han evolucionado en condiciones de baja salinidad del suelo y no muestran tolerancia a la sal [23].

La salinidad es uno de los factores más graves que limitan la productividad de los cultivos agrícolas, con efectos adversos sobre la germinación, el vigor de la planta y el rendimiento de los cultivos [24].

En todo el mundo, más de 45 millones de hectáreas de tierras de regadío han sido dañadas por la sal, y 1,5 millones de hectáreas se retiran de la producción cada año como resultado de los altos niveles de salinidad en
el suelo [24].

La alta salinidad afecta a las plantas de varias formas; reducen el crecimiento, el desarrollo y la supervivencia de las plantas. Durante el inicio y desarrollo del estrés salino dentro de una planta, todos los procesos principales, como como la fotosíntesis, la síntesis de proteínas y el metabolismo energético y lipídico se ven afectados [19, 25, 26, 27]. Durante la exposición inicial a la salinidad, las plantas experimentan estrés hídrico, que a su vez reduce la expansión de la hoja. Los efectos osmóticos del estrés por salinidad se pueden observar inmediatamente después de la aplicación de sal y se cree que continúan

durante la duración de la exposición, lo que resulta en la inhibición de la expansión celular y la división celular, así como el cierre de las estomas [19].

Durante la exposición prolongada a la salinidad, las plantas experimentan estrés iónico, que puede conducir una reducción del área disponible para apoyar el crecimiento continuo [28]. De hecho, el exceso de sodio y lo que es más importante, el cloruro de sodio tiene el potencial de afectar las enzimas de las plantas y causar hinchazón, lo que resulta en una reducción de la producción de energía y otros cambios fisiológicos [29].

2.3 Problemática de la salinidad en el valle del Guadalhorce y solución actual

La red hidrográfica de la región del Trías de Antequera, donde se sitúa el valle del Guadalhorce, está constituida, fundamentalmente, por el río Guadalhorce que recorre la zona de este a oeste (figura 2). En el tramo final de su cuenca alta, antes de atravesar el Tajo de los Gaitanes, recibe las aportaciones de los ríos Guadalteba y Turón, procedentes de la Serranía de Ronda. En el entorno de la unión de los tres ríos se construyó en primer lugar la presa del Conde del Guadalhorce sobre el río Turón y posteriormente las presas del Guadalhorce y del Guadalteba en cada uno de estos ríos que en aguas altas forman un solo embalse. A este conjunto se le denomina “Los embalses del Guadalhorce”, cuyas aguas se utilizan para regadíos y para el abastecimiento ciudad de Málaga.

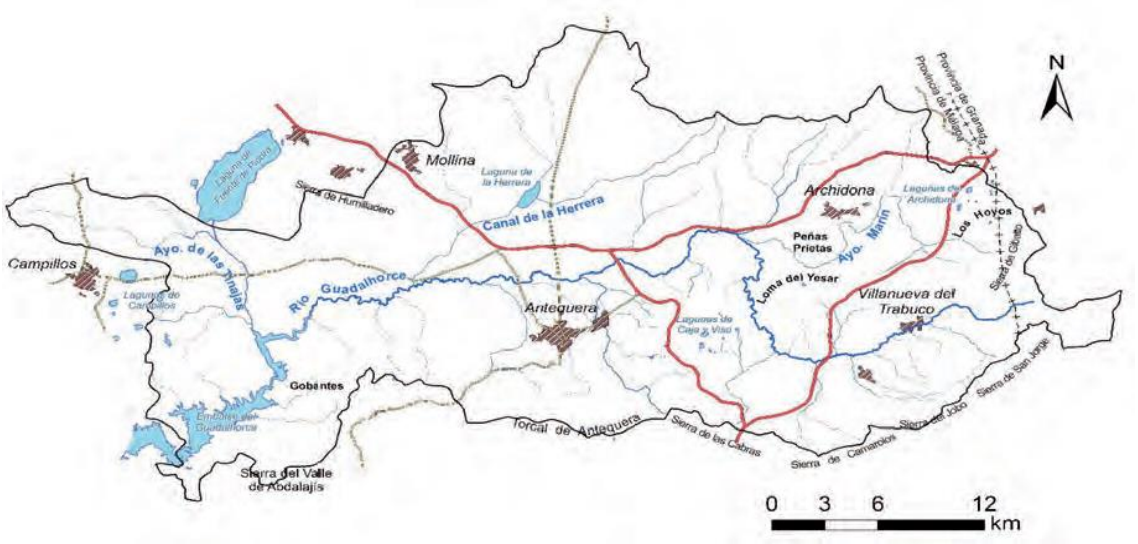


Figura 2: Hidrografía de la región del Trías de Antequera

Las primeras investigaciones acerca de la calidad de las aguas del río Guadalhorce realizadas en las décadas de 1960 y 1970 mostraron la alta salinidad que presentaban las aguas del río en una estación de aforos cercana a la zona de Gobantes, que posteriormente quedó inundada por el embalse [30, 31].

Por entonces, mediante muestreos en una serie de puntos del río desde su nacimiento hasta el embalse del Guadalhorce, se pusieron de manifiesto las variaciones de salinidad de las aguas de escorrentía y su relación con los materiales de la cuenca vertiente. Se constató que el mayor aumento de salinidad se producía en el estrecho de Meliones, debido a la existencia de un manantial muy salino [32, 33].

Por lo tanto, se ha constatado que son varios los manantiales salinos que desembocan en el embalse del Guadalhorce. Estos manantiales se conocen como Manantiales de Meliones, pero existen algunos individualizados con sus nombres propios que desembocan en puntos distintos del embalse e incluso lejos del embalse (Manantial del Cañaveralejo) pero que desaguan finalmente en él. Esta definición conjunta se justifica en cierto sentido porque cuando se ha intentado captar las aguas de un manantial determinado o cerrar determinados caminos para la afluencia de aguas hipersalinas definidas al embalse, las surgencias han aparecido en otro punto mostrando la evidencia de que nos enfrentamos a un karst que actúa como una sola unidad hidrológica y que desagua aguas hipersalinas de calidad y cantidad variables en función de las incidencias meteorológicas [34].

Cualitativamente podríamos decir que el comportamiento conjunto de los elementos productores de salinidad es como sigue: debe existir un domo salino o karst que proporciona iones a las aguas que están en contacto con él y, también, a las que circulan con una cierta velocidad en contacto con él. Por tanto, en época de aguas bajas circulan caudales menores pero muy salinos (aguas que han tenido un tiempo mayor de contacto con el karst). Pero en época de lluvias también se producen aumentos de salinidad, pues los mayores flujos, con mayor turbulencia, deben aumentar el traspaso de iones de la roca al agua. Por todo ello, es difícil estudiar en detalle el macizo salino y su hidrología asociada, como hemos señalado [30, 31, 34].

La dificultad en la lucha contra estas surgencias reside precisamente en ese comportamiento zonal más que una serie de flujos lineales concretos de las aguas que llegan al embalse. En condiciones naturales, los puntos de llegada de las aguas salobres al río debieron estar más definidos concentrándose en los puntos más bajos y de permeabilidad preferencial del macizo origen del flujo en contacto con los sedimentos menos permeables del cauce del río, pero con la subida del nivel de las aguas en el embalse, esos puntos de desagüe originales fueron inundados y los vertidos hipersalinos cambiaron de localización, siendo unos subacuáticos (que vierten por debajo del nivel de las aguas embalsadas) y otros subaéreos (que vierten aún por encima de las aguas embalsadas). Pero esas localizaciones, sobre todo las subacuáticas, no son estables [34].

La necesidad obliga a utilizar procesos manuales de mezclas de agua, pero la cantidad procedente del embalse del Guadalhorce no puede ser muy alta. El resultado es doble: la calidad del agua obtenida es de baja calidad, si la mezcla no se realiza adecuadamente, pues se pretende usar toda la que es posible extraer del Guadalhorce con la limitación impuesta por el límite de salinidad

compatible con los riegos; por otra parte, el nivel del agua en el embalse del Guadalhorce, del que se extrae menos agua, sube incesantemente en épocas de lluvias, amenazando con llegar al nivel del collado por el que las aguas salobres se habrían mezclado con las del embalse de Guadalteba [34].

Pero tampoco es fácil desprenderse de estas aguas sobrantes del embalse del Guadalhorce (no son sobrantes porque hacen falta para el consumo, sino que son difícilmente utilizables por su salinidad). Si los vertidos de esas aguas al cauce fueran frecuentes, la ecología completa del valle del Río Guadalhorce sería afectada (figura 3). Por otra parte, como las aguas a utilizar se toman del propio río, aguas debajo de las presas, hay que usar ventanas de oportunidad (épocas de lluvia en que las aguas se diluye parcialmente y en la que no se riega) para aprovechar y lanzar al mar esas aguas pretendidamente sobrantes [34].

Por lo tanto, es necesario encontrar cual es una mezcla de agua óptima, esto es utilizar el máximo posible de agua de alta salinidad del embalse del Guadalhorce garantizando que la mezcla sigue siendo utilizable.

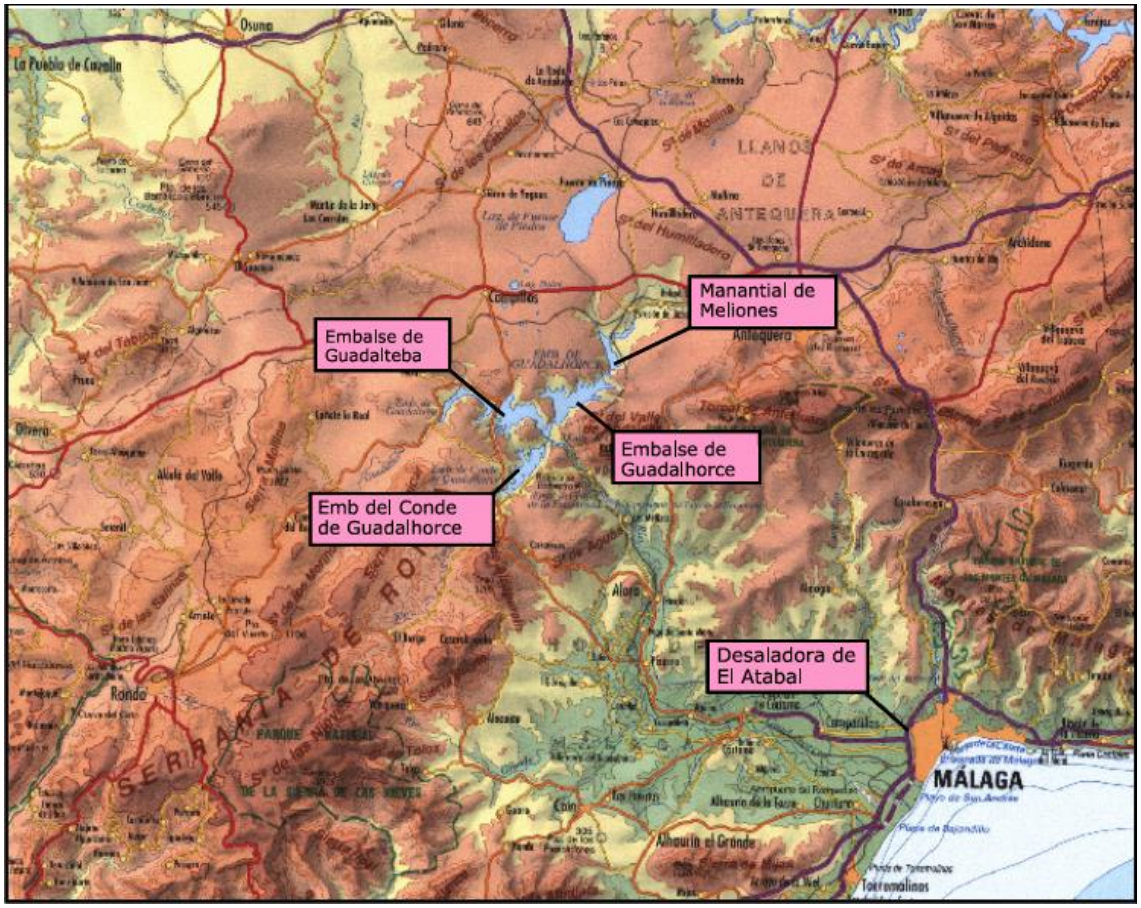


Figura 3: Valle del Guadalhorce y sistema de presas

2.4 Aplicación de métodos de aprendizaje automático a la calidad del agua

En este apartado se analizan los distintos estudios en los que se han utilizado métodos de aprendizaje automático y/o estadísticos para la predicción de distintos parámetros de calidad del agua de reservas o presas, con la idea de comprender el provecho de aplicar este tipo de tecnologías y métodos en beneficio de un mejor uso del agua.

El desarrollo de modelos precisos y fiables puede resultar valioso para superar el problema en la gestión del agua utilizada en la agricultura. Con este pretexto, la facultad de ciencias de Casablanca, Marruecos, llevó a cabo un estudio en una serie de presas al sur de Rabat, donde confluyen los ríos Kourifla, Machraa, Grou and Bouregreg. Se utilizaron 8 modelos de aprendizaje automático (ML), a saber: red neuronal artificial (ANN), regresión lineal múltiple (MLR), árbol de decisión, random forest (RF), máquinas de soporte vectorial (SVM), k-vecinos más cercanos (kNN), descenso de gradiente estocástico (SGD) y Adaptive Boosting (AdaBoost), para predecir 10 parámetros de calidad del agua de riego (IWQ, por sus siglas en inglés), entre ellos el porcentaje de sodio (%Na), ratio de absorción de sodio (RAS, o SAR en inglés) y total de sales disueltas (TDS en inglés), entre otros. Los resultados revelarón que, a excepción de los modelos SVM y k-NN, todos los demás modelos son altamente precisos para predecir parámetros con coeficientes de correlación (r) con rangos de [0.56, 0.99] y [0.64, 0.99] para entrenamiento y procesos de validación respectivamente [35].

Existen estudios que toman datos recolectados durante un largo periodo de tiempo. Este es el caso del estudio de Chou et al. (2018), donde se valen de datos recolectados durante 10 años (1995-2016) de 20 reservas de Taiwan para entrenar distintos modelos de aprendizaje automático (redes neuronales, maquinas de soporte vectorial, arboles de decisión y regresión y regresiones lineales), con el objetivo de crear modelos predictivos de parámetros de calidad del agua [36].

Otros estudios defienden el uso de métodos de aprendizaje automático como forma de ahorrar costes y tiempo en los análisis de la calidad de el agua. Sostienen que, dadas las condiciones de algunos países, se hace imprescindible dotar a las presas o reservas de agua de mecanismos flexibles que agilicen la toma de decisiones. Con esta motivación, Ahmed et al. exploran distintos métodos de aprendizaje automático supervisado para estimar un parámetro de calidad de agua que categoriza dicha calidad en 5 clases (muy mala, mala, media, buena, excelente) [37, 38]. Como resultado se consigue clasificar el mencionado parámetro de calidad del agua con una precisión del 85% con redes neuronales. Para concluir, esta investigación propone la incorporación de este tipo de algoritmos a sensores de IoT para lograr el análisis y predicción de la calidad del agua en tiempo real, como sistemas de monitorización [39].

Existen estudios que usan imágenes de muestras de aguas en lugar de variables. Es el caso del estudio de Ankit y Elliott (2019), donde se presenta un modelo de redes neuronales convolucionales (CNN por sus siglas en inglés) que es entrenado con un conjunto de 105 imágenes etiquetadas correspondientes con distintos niveles de contaminación. Este modelo consigue una precisión de clasificación del 96%. La investigación propone desplegar este modelo como una forma eficiente de asegurar que los niveles de contaminación del agua son correctos, y de no ser así rápidamente detectar dicha contaminación y poner en alerta al gobierno local o nacional de cara a tomar medidas para asegurar la salud de la población [40].
Siguiendo con las redes neuronales, Nabeel et al. (2012) aplica una red neuronal artificial (ANN por sus siglas en inglés) *feed-forward* (alimentada hacia adelante), completamente conectada y de tres capas para la predicción de 36 parámetros de calidad del agua (WQI) del río Kinta, en Malasia. Entre estos parámetros se encuentran temperatura, conductividad, salinidad, sodio, TDS, etc. El resultado es que las predicciones de este modelo pueden explicar el 95,4% de las variaciones en los parámetros mencionados anteriormente. El enfoque presentado en este artículo ofrece una alternativa útil y poderosa al cálculo y predicción tradicional de los índices de calidad del agua, especialmente en el caso de métodos de cálculo de dichos índices implican cálculos largos y uso de varias fórmulas para cada valor, o rango de valores, de las variables constituyentes de la calidad del agua [41].

Mohamad Sakizadeh (2016) inició un estudio para predecir índices de calidad del agua (WQI) utilizando redes neuronales artificiales (ANN) con respecto a 16 variables calidad del agua subterránea recolectadas de 47 pozos y manantiales en Andimeshk (Irán) durante 2006-2013 por el Ministerio de Energía de Irán. Tal predicción tiene el potencial de reducir el cálculo tiempo, esfuerzo y la posibilidad de error en el cálculo. Para ello, se utilizaron tres algoritmos de ANN, incluyendo ANN con parada anticipada, ANN conjuntas y ANN con regularización bayesiana. Los coeficientes de correlación entre las predicciones y las observaciones de los valores de WQI fueron 0.94 y 0.77 para los conjuntos de datos de prueba y entrenamiento, respectivamente [42].

Como se puede ver, la aplicación de mecanismos de aprendizaje automático para la predicción de parámetros de calidad del agua, ya sea en presas, reservas, cuencas o ríos, es un enfoque recurrente en los últimos años (se puede ver que todos los estudios mencionados son a partir de 2010, como mínimo) dado el ahorro de costes, flexibilidad y rapidez que puede suponer a la hora de la toma de decisiones.

En este documento se recurrirán a mecanismos de aprendizaje automático para la predicción de parámetros de calidad del agua de las presas del valle del Guadalhorce, en particular, se pretenderá hallar un modelo que clasifique las muestras de agua según el su índice de cloruro sódico (NaCl) con el objetivo de hallar las variables que influyen en una mezcla óptima de agua.

2.5 Preguntas a responder

En base a lo expuesto anteriormente algunas preguntas a responder durante el desarrollo del trabajo son las siguientes:

¿Es posible hallar una mezcla óptima de agua usable maximizando la cantidad de agua de la presa Guadalhorce? Si es posible, ¿Qué parámetros son necesarios para lograr esta mezcla óptima?

¿Cómo se podría mejorar la toma de datos en el sistema de presas?

¿Es posible hallar una predicción de la salinidad teniendo en cuenta los datos de los que disponemos?

¿Qué parámetros, de los que tenemos datos, influyen mayormente en la salinidad de la presa del Guadalhorce?

¿Qué algoritmo o algoritmos presentan mejor capacidad predictiva de la salinidad presente en la presa?

¿Qué beneficios aportaría la aplicación de mecanismos de aprendizaje automático a la gestión de la salinidad en el sistema de presas del Valle del Guadalhorce?

3. Proceso de implementación

Siguiendo la literatura tradicional de la gestión de datos, estos son generados o capturados, almacenados, preprocesados, analizados, visualizados y publicados, de manera que se cierra el círculo y se permite su reutilización. Cada una de estas fases tiene un objetivo, generando valor a partir de los datos en cada una de ellos. Como ocurre en el desarrollo de este trabajo, algunas fases se solapan, así, por ejemplo, la etapa de visualización no solo se realizará posteriormente al análisis, si no, antes y durante el mismo. De esta forma, se entiende que las fases típicas del ciclo de vida de los datos son las siguientes [43]:

- Captura
- Almacenamiento
- Preprocesado
- Análisis
- Visualización
- Publicación

La finalidad principal de aplicar el ciclo de vida mencionado anteriormente a los datos proporcionados por la Junta de Andalucía no es otra que la extracción de conocimiento.

Teniendo como referencia la extracción de conocimiento a partir de los datos proporcionados, se puede recurrir a la figura 4 para representar esta finalidad. Es lo que se conoce como la pirámide DIKW (*data, information, knowledge y wisdom*), de modo que la información se define a partir de los datos disponibles, el conocimiento se extrae de dicha información y la sabiduría es entendida como la habilidad para aplicar dicho conocimiento en beneficio propio o común [44].

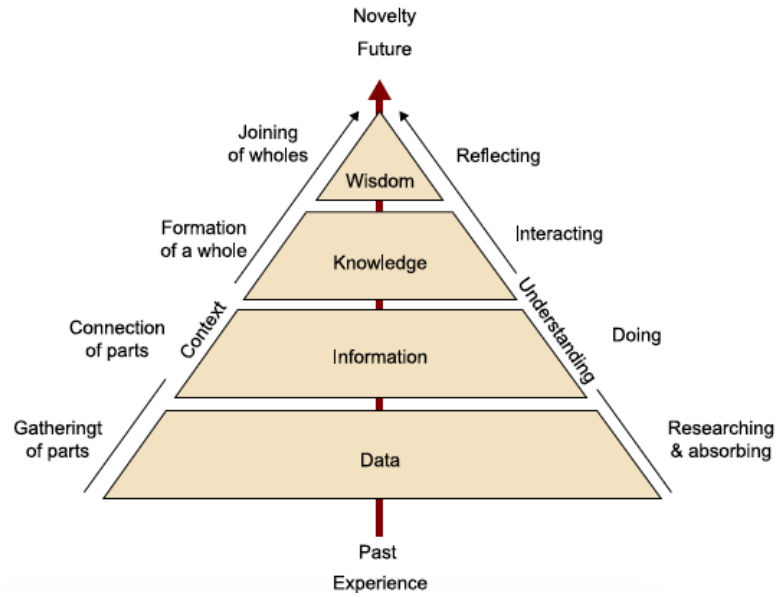


Figura 4: Pirámide DIKW

3.1 Captura de datos

Los datos proporcionados por la Junta de Andalucía contienen información de distintos parámetros de explotación del sistema de presas, que serán descritos en el próximo apartado, de 2012 a 2019.

La Junta de Andalucía dispone de datos de explotación de la presa de hace más de 40 años, sin embargo, por trabajar con un formato homogéneo, se tomarán los datos de explotación desde 2012 a 2019, ambos años inclusive.

La captura de estos datos ha sido históricamente un proceso manual, incluso hoy en día lo sigue siendo, aunque se han incorporado otros mecanismos de captura automática que se comentarán mas adelante.

Diariamente, un técnico de la Junta de Andalucía visita distintos lugares de la explotación y realiza toma de datos provenientes de distintos sistemas y fuentes:

-Toma de datos en la caseta de meteorología situada en el Conde de Guadalhorce a las 8:00 horas. Se anotan:

- Temperatura máxima registrada el día anterior.
- Temperatura mínima registrada el día anterior.
- Precipitaciones (en mm) registradas en 24 horas (de 8:00 a 8:00)
- Evaporación (en mm). Se obtiene de la diferencia con respecto al día anterior.

-Toma de cotas embalses:

- Forma visual en escala.
- SAIH. De cada embalse, se comprueba la variación en centímetros 24 horas y se anota esa diferencia. El SAIH es una red de estaciones remotas distribuidas en toda la superficie de la Demarcación Hidrográfica de las Cuencas Mediterráneas Andaluzas, para obtener, a tiempo real, información de las incidencias hidrometeorológicas [45].
- Actualmente, a diario se realizan ambas operaciones de manera sistemática y por corroborar datos.

Anotaciones de desembalses:

- Guadalhorce-Guadalteba: al disponer de caudalímetros, se comprueba la diferencia de m3 que hay en el archivo histórico de la aplicación. Siempre que el caudalímetro funcione, se calcula de forma teórica.
- Conde de Guadalhorce: no se tiene controlado por un caudalímetro. Sus desembalses son siempre estimados, ya sean por central hidroeléctrica o por desagüe de fondo.

-Aportación aparente:

- Es el resultado de sumar la variación del embalse con el volumen total desembalsado (filtraciones y desembalses). Es un valor que puede ser positivo o negativo.

-Volumen evaporado:

- Es una operación en la que, según el valor de la evaporación obtenido en caseta de meteorología, estima los m3 evaporados según la cota (y su correspondiente volumen) de cada embalse.

-Estadillo "Aguas abajo":

- Cada mañana, ENEL, facilita los datos de "sobrante" en Tajo de la Encantada y cota de Gaitanejo. Eso da como resultado un volumen total del que podremos usar para satisfacer demandas.

- Se anota el caudal que va para abastecimiento, riegos y caudal ecológico. Lo que arroja un sumatorio de volumen diario.

3.2 Almacenamiento y descripción del conjunto de datos

Los datos diariamente capturados o recolectados manualmente, se recogen en un fichero con extensión ".xls". Este fichero ".xls", conocido como estadillo, termino usado a partir de ahora, tiene un formato muy particular, diseñado para ser impreso y ser presentado como un informe mensual, como puede verse en la figura 5.

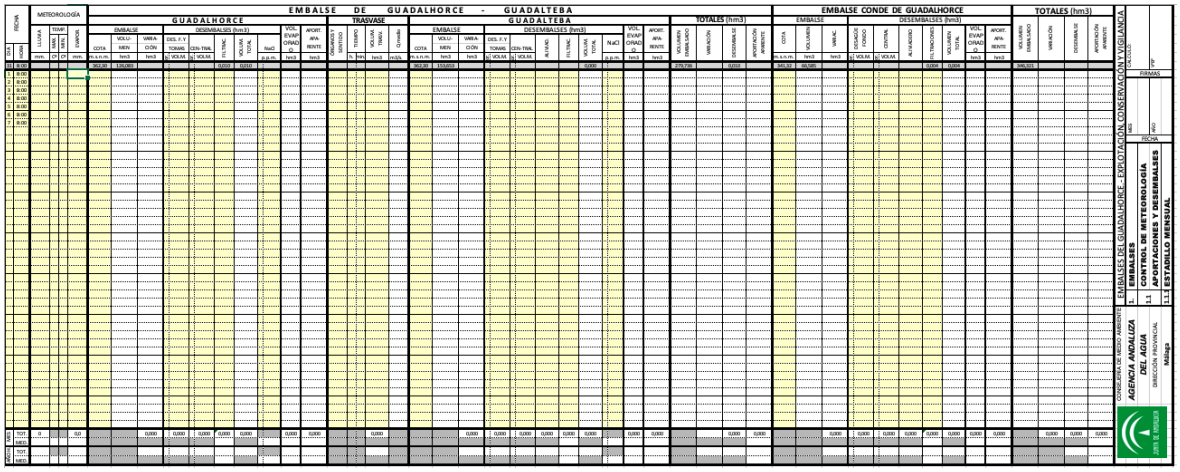


Figura 5: Estadillo mensual Agencia Andaluza del Agua 2012

Desde el punto de vista de la ciencia de datos, cabe reseñar que el almacenamiento de datos en ficheros de softwares propietarios, cuyo uso requiere una licencia, no siempre es lo mas recomendable, como se puede ver en el modelo representado en la figura 6, conocido como esquema de 5 estrellas para la publicación de datos, creado por Tim Berners-Lee [46].

Figura 6: Esquema de 5 estrellas para la publicación de los datos

Cada estadillo, como el de la figura 5, contiene información de explotación mensual del sistema de presas del valle del Guadalhorce. A grandes rasgos, este conjunto de datos almacena; información meteorológica del enclave donde se sitúan el conjunto de presas, volumen de agua almacenada por cada presa individual y por conjunto de presas, volumen de agua desembalsada para satisfacer las distintas demandas (principalmente consumo, riego y vertidos al cauce del río) y, por último, volumen de agua destinada para los distintos consumos, comentados anteriormente.

Si omitimos las cabeceras del estadillo, cada fila del estadillo se corresponde con los datos recogidos de distintas variables en un día, por lo tanto, las filas de cada estadillo variarán entre 28 y 31, dependiendo del mes en el que se hayan recogido los datos.

De igual forma, si no consideramos las cabeceras, cada estadillo contendría 71 variables o columnas. Cada columna representa un valor especifico de explotación del sistema de presas. Adicionalmente a las 71 variables contenidas en el estadillo, se introducen manualmente dos variables auxiliares, correspondiente con el mes y año de cada toma diaria.

Como se indicó anteriormente, de los datos que nos facilita la Junta de Andalucía, por homogeneidad en el formato, seleccionamos los estadillos comprendidos entre 2012 y 2019, ambos inclusive, para realizar el estudio. Por lo tanto, si cada estadillo contiene datos de explotación relativos a un mes, tendremos un total de 96 estadillos. La integración del conjunto de estadillo es objeto del siguiente apartado.

A continuación, se describirán las variables presentes en los distintos estadillos mensuales, cada cual representará un parámetro de explotación del sistema de presas:

- *Dia*: Se corresponde con el día en el que se produce la toma de datos. Esta variable tomará un valor de 0 a 31, dependiendo del mes donde se realice la toma

- *Hora*: Esta variable almacena la hora en la que se han tomado los datos. Toma un único valor, que es 8:00. Dado que este valor no varía, no será una variable de interés en este estudio.

- *Mes:* Se trata de una variable creada manualmente que almacena el mes donde se ha tomado cada dato. Toma valores de 1 a 12, según los meses correspondiente a un año. El fin de introducir esta variable, como se verá posteriormente, es crear un campo fecha con formato dia-mes-año, para identificar unívocamente a cada registro y poder aplicar análisis de series temporales.

- *Anyo:* Se trata de una variable creada manualmente que almacena el año donde se ha tomado cada dato. Toma valores de 2012 a 2019, según los años que estamos evaluando. El fin de introducir esta variable, como se verá posteriormente, es crear un campo fecha con formato dia-

mes-año, para identificar unívocamente a cada registro y poder aplicar análisis de series temporales.

- *Lluvia:* Variable que contiene el valor de las precipitaciones que se han producido en un día. Su valor viene dado en mm y se trata de una variable numérica de tipo float, ya que puede tomar valores decimales. El valor de esta variable se recoge en el SAIH.

- *Temperatura_Max:* En esta variable se almacena la temperatura máxima, en Cº, recogida por el SAIH. Se trata de una variable numérica de tipo float, ya que pueden recogerse valores decimales.

- *Temperatura_Min:* En esta variable se almacena la temperatura mínima, en Cº, recogida por el SAIH. Se trata de una variable numérica de tipo float, ya que pueden recogerse valores decimales.

- *Evaporacion:* Almacena la lectura diaria de evaporación del tanque situado en la estación meteorológica. Su valor viene dado en mm y se trata de una variable numérica de tipo float.

- *Cota_guadalhorce:* Variable que almacena la altura sobre el nivel del mar de la presa Guadalhorce. Esta variable viene dada en m y se trata de una variable numérica de tipo float.

- *Volumen_embalsado_guadalhorce:* Contiene los datos diarios sobre el volumen de agua contenido en el embalse Guadalhorce. Este valor se obtiene a través de caudalímetros y su magnitud viene dada en hm^3. Se trata de una variable numérica de tipo float, por la existencia de valores decimales.

- *Variacion_volumen_guadalhorce:* Esta variable contiene la variación de volumen de la presa Guadalhorce entre el día x (día de la toma de datos) y x-1 (día anterior al día de la toma de datos). Puede toma un valor positivo o negativo, ya que la variación respecto al día anterior puede ser positiva o negativa. Su magnitud viene dada en hm^3. Se trata de una variable numérica de tipo float.

- *Tiempo_desembalse_ guadalhorce:* Contiene información acerca del tiempo durante el cual se ha realizado un desembalse de la presa Guadalhorce. Su magnitud viene dada en horas. Esta variable contiene el carácter ":", por lo que habrá que tenerlo en cuenta cuando se realice la lectura.

- *Caudal_desembalse_guadalhorce:* Esta variable almacena el caudal diario de desembalse de la presa Guadalhorce. Su magnitud viene dada en m^3/s. Se trata de una variable numérica de tipo float

- *Filtracion_guadalhorce:* Muestra el volumen de agua que se pierde por filtraciones en la presa Guadalhorce. Su magnitud es hm^3 y toma un valor constante de 0,02.

- *Volumen_desembalse_guadalhorce:* Esta variable muestra el volumen total diario de agua desembalsada en la presa Guadalhorce por la central hidroeléctrica, a la que sumamos los desagües y filtraciones de agua, comentados anteriormente. La magnitud de esta variable es hm^3 y se trata de una variable numérica de tipo float.

- *Salinidad_guadalhorce:* Esta variable contiene datos diarios sobre la salinidad o NaCl presente en el embalse Guadalhorce, medida en p.p.m. Se trata de una variable numérica de tipo float.

- *Toneladas_sal_guadalhorce:* Esta variable, al igual que la anterior contienen valores de salinidad, en este caso, medida en toneladas. También, se trata de una variable numérica de tipo float.

- *Volumen_evaporado_guadalhorce:* Muestra la estimación del volumen total evaporado en un día en la presa del Guadalhorce. Es un valor obtenido a través de la casetilla de meteorología del sistema de explotación. Su magnitud es hm^3 y se trata de una variable numérica de tipo float.

- *Aportacion_aparente_guadalhorce:* Esta variable se halla a partir de la suma de la variación del volumen respecto al día anterior, valor almacenado en *Variacion_volumen_guadalhorce*, y el volumen que se ha desembalsado, guardado en la variable *Volumen_desembalse_guadalhorce.* Su magnitud es hm^3 y es una variable numérica de tipo float.

- *Cota_guadalteba:* Variable que almacena la altura sobre el nivel del mar de la presa Guadalteba. Esta variable viene dada en m y se trata de una variable numérica de tipo float.

- *Volumen_embalsado_guadalteba:* Contiene los datos diarios sobre el volumen de agua contenido en el embalse Guadalteba. Este valor se obtiene a través de caudalímetros y su magnitud viene dada en hm^3. Se trata de una variable numérica de tipo float, por la existencia de valores decimales.

- *Variacion_volumen_guadalteba:* Esta variable contiene la variación de volumen de la presa guadalteba entre el día x (día de la toma de datos) y x-1 (día anterior al día de la toma de datos). Puede toma un valor positivo o negativo, ya que la variación respecto al día anterior puede ser positiva o negativa. Su magnitud viene dada en hm^3. Se trata de una variable numérica de tipo float.

- *Tiempo_desembalse_guadalteba:* Contiene información acerca del tiempo durante el cual se ha realizado un desembalse de la presa Guadalteba. Su magnitud viene dada en horas. Esta variable contiene el carácter ":", por lo que habrá que tenerlo en cuenta cuando se realice la lectura.

- *Caudal_desembalse_guadalteba:* Esta variable almacena el caudal diario de desembalse de la presa Guadalteba. Su magnitud viene dada en m^3/s. Se trata de una variable numérica de tipo float.

- *Filtraciones_guadalteba:* Muestra el volumen de agua que se pierde por filtraciones en la presa Guadalteba, sin embargo, no contiene datos.

- *Volumen_desembalse_guadalteba:* Esta variable muestra el volumen total diario de agua desembalsada en la presa Guadalteba por la central hidroeléctrica, a la que sumamos los desagües y filtraciones de agua, comentados anteriormente. La magnitud de esta variable es hm^3 y se trata de una variable numérica de tipo float.

- *Salinidad_guadalteba:* Esta variable contiene datos diarios sobre la salinidad o NaCl presente en el embalse Guadalteba, medida en p.p.m. Se trata de una variable numérica de tipo float.

- *Toneladas_sal_guadalteba:* Esta variable, al igual que la anterior contienen valores de salinidad, en este caso, medida en toneladas. También, se trata de una variable numérica de tipo float.

- *Volumen_evaporado_guadalteba:* Muestra la estimación del volumen total evaporado en un dáa en la presa Guadalteba. Es un valor obtenido a través de la casetilla de meteorología del sistema de explotación. Su magnitud es hm^3 y se trata de una variable numérica de tipo float.

- *Aportacion_aparente_guadalteba:* Esta variable se halla a partir de la suma de la variación del volumen respecto al día anterior, valor almacenado en *Variacion_volumen_guadalteba*, y el volumen que se ha desembalsado, guardado en la variable *Volumen_desembalse_guadalteba.* Su magnitud es hm^3 y es una variable numérica de tipo float.

- *Volumen_total_guadalhorce_guadalteba:* Esta variable contiene la suma de los volúmenes embalsados de las presas Guadalhorce y Guadalteba. Su magnitud es hm^3 y es una variable numérica de tipo float.

- *Variacion_total_guadalhorce_guadalteba:* Esta variable contiene la suma de la variación diaria de las presas Guadalhorce y Guadalteba. Su magnitud es hm^3 y es una variable numérica de tipo float.

- *Desembalse_total_guadalhorce_guadalteba:* Esta variable contiene la suma de los desembalses de las presas Guadalhorce y Guadalteba. Es una variable importante, ya que los desembalses de ambas presas son

usados para hallar una mezcla de agua que poder utilizar para regadíos y suministro de agua para consumo. Su magnitud es hm^3 y es una variable numérica de tipo float.

- *Aportacion_aparente_total_guadalhorce_guadalteba:* Esta variable contiene la suma de la aportación aparente de las presas Guadalhorce y Guadalteba. Su magnitud es hm^3 y es una variable numérica de tipo float.

- *Cota_conde:* Variable que almacena la altura sobre el nivel del mar de la presa Conde de Guadalhorce. Esta variable viene dada en m y se trata de una variable numérica de tipo float.

- *Volumen_embalsado_conde:* Contiene los datos diarios sobre el volumen de agua contenido en el embalse Conde del Guadalhorce. Este valor se obtiene a través de caudalímetros y su magnitud viene dada en hm^3. Se trata de una variable numérica de tipo float, por la existencia de valores decimales.

- *Variación_volumen_conde:* Esta variable contiene la variación de volumen de la presa Conde del Guadalhorce entre el día x (día de la toma de datos) y x-1 (día anterior al día de la toma de datos). Puede toma un valor positivo o negativo, ya que la variación respecto al día anterior puede ser positiva o negativa. Su magnitud viene dada en hm^3. Se trata de una variable numérica de tipo float.

- *Filtraciones_conde:* Muestra el volumen de agua que se pierde por filtraciones en la presa Conde del Guadalhorce. Su magnitud es hm^3 y se trata de una variable numérica de tipo float.

- *Volumen_desembalse_conde:* Esta variable muestra el volumen total diario de agua desembalsada en la presa Conde del Guadalhorce por la central hidroeléctrica, a la que sumamos los desagües y filtraciones de agua, comentados anteriormente. La magnitud de esta variable es hm^3 y se trata de una variable numérica de tipo float.

- *Salinidad_conde:* Esta variable pretende contener datos de salinidad de la presa Conde del Guadalhorce en unidades p.p.m, aunque como se verá en el próximo apartado, no contiene información útil.

- *Toneladas_sal_conde:* Esta variable pretende contener datos de salinidad de la presa Conde del Guadalhorce en unidades toneladas, aunque como se verá en el próximo apartado, no contiene información útil.

- *Volumen_evaporado_conde:* Muestra la estimación del volumen total evaporado en un dia en la presa Conde del Guadalhorce. Es un valor obtenido a través de la casetilla de meteorología del sistema de explotación. Su magnitud es hm^3 y se trata de una variable numérica de tipo float.

- *Aportacion_aparente_conde:* Esta variable se halla a partir de la suma de la variación del volumen respecto al día anterior, valor almacenado en *Variacion_volumen_conde*, y el volumen que se ha desembalsado, guardado en la variable *Volumen_desembalse_conde.* Su magnitud es hm^3 y es una variable numérica de tipo float.

- *Volumen_total_guadalhorce_guadalteba_conde (fallo ortográfico en xls):* Esta variable contiene la suma de los volúmenes embalsados de las presas Guadalhorce, Guadalteba y Conde de Guadalhorce*.* Su magnitud es hm^3 y es una variable numérica de tipo float.

- *Variación_total_guadalhorce_guadalteba_conde:* Esta variable contiene la suma de la variación diaria de las presas Guadalhorce, Guadalteba y Conde de Guadalhorce*.* Su magnitud es hm^3 y es una variable numérica de tipo float.

- *Desembalse_total_guadalhorce_guadalteba_conde:* Esta variable contiene la suma de los desembalses de las presas Guadalhorce, Guadalteba y Conde de Guadalhorce. Su magnitud es hm^3 y es una variable numérica de tipo float.

- *Aportación_aparente_total_guadalhorce_guadalteba_conde:* Esta variable contiene la suma de la aportación aparente de las presas Guadalhorce, Guadalteba y Conde de Guadalhorce*.* Su magnitud es hm^3 y es una variable numérica de tipo float.

- *Caudal_total_demandas:* Esta variable contiene datos sobre el caudal que hay que suministrar para los diferentes usos (riego y consumo) y para el caudal mínimo que lleva el río por razones ecológicas. Su magnitud es m^3/s y se trata de una variable numérica de tipo float.

- *Volumen_total_demandas:* Contiene datos sobre el volumen diario necesario para satisfacer las distintas demandas (riego, consumo y vertidos de agua al caudal del rio por motivos ecológicos). Esta variable se mide en hm^3 y se trata de una variable numérica de tipo float.

- *Salinidad_mezcla:* Resultante de mezclar las aguas de las distintas presas; mayormente de las presas de Guadalhorce, Guadalteba, se obtiene un agua para cubrir distintas demandas (riego, consumo y vertidos de agua al caudal del rio por motivos ecológicos). Esta variable se usará como variable etiquetada en el dataset. Esta variable contiene la salinidad de dicha mezcla de aguas. Se mide en p.p.m y se trata de una variable numérica de tipo float.

- *Toneladas_sal_mezcla:* Se trata de la misma variable que *Salinidad_mezcla* pero medida en otra magnitud, en este caso, toneladas.

- *Cota_Gaitanejo:* Después de las tres presas grandes hay una pequeña presa que funciona como cámara de carga de la central de nuevo chorro. Esta variable viene dada en m y se trata de una variable numérica de tipo float.

- *Volumen_gaitanejo:* Muestra el volumen contenido en la pequeña presa de Gaitanejo. Su magnitud es hm^3 y es una variable numérica de tipo float.

- *Volumen_central_reversible_tajo:* La central reversible del Tajo de La Encantada tiene un depósito superior de más de 3,5 hm3, que es el volumen que la compañía eléctrica puede bombear y luego turbinar en función de sus necesidades. Lo que supera a los 3,5 es el agua se tiene para abastecer las demandas. Su magnitud es hm^3 y se trata de una variable numérica de tipo float.

- *Volumen_total_gaitanejo_tajo:* Suma de los volúmenes de las presas pequeñas Gaitanejo y Tajo La Encantada. La magnitud de esta variable es hm^3 y se trata de una variable numérica de tipo float.

- *Variacion_total_gaitanejo_tajo:* Suma de las variaciones de las presas pequeñas Gaitanejo y Tajo La Encantada. La magnitud de esta variable es hm^3 y se trata de una variable numérica de tipo float.

- *Caudal_abastecimiento:* Contiene datos sobre el caudal de agua destinado al abastecimiento o consumo. Se mide en m^3/s y se trata de una variable numérica de tipo float.

- *Horas_abastecimiento:* Muestra el tiempo durante el cual se ha usado el agua para fines de abastecimiento o consumo. Esta variable se mide en horas y se trata de una variable numérica de tipo entero.

- *Min_abastecimiento:* Muestra el tiempo durante el cual se ha usado el agua para fines de abastecimiento o consumo. Esta variable se mide en minutos y se trata de una variable numérica de tipo entero.

- *Volumen_abastecimiento:* Contiene datos sobre el volumen de agua destinado al abastecimiento o consumo. Se mide en hm^3 y se trata de una variable numérica de tipo float.

- *Caudal_riego:* Contiene datos sobre el caudal de agua destinado a fines de riego Se mide en m^3/s y se trata de una variable numérica de tipo float.

- *Horas_riego:* Muestra el tiempo durante el cual se ha usado el agua para fines de riego. Esta variable se mide en horas y se trata de una variable numérica de tipo entero.

- *Min_riego:* Muestra el tiempo durante el cual se ha usado el agua para fines de riego. Esta variable se mide en minutos y se trata de una variable numérica de tipo entero.

- *Volumen_riego:* Contiene datos sobre el volumen de agua destinado al riego. Se mide en hm^3 y se trata de una variable numérica de tipo float.

- *Caudal_ecologico:* Contiene datos sobre el caudal de agua destinado a caudal ecológico. Esto es la cantidad de agua que se devuelve al cauce del río, por cuestiones ecológicas. Se mide en m^3/s y se trata de una variable numérica de tipo float.

- *Horas_ecologico:* Muestra el tiempo durante el cual se ha usado el agua para ecológicos. Esta variable se mide en horas y se trata de una variable numérica de tipo entero.

- *Min_ecologico:* Muestra el tiempo durante el cual se ha usado el agua para fines ecológicos. Esta variable se mide en minutos y se trata de una variable numérica de tipo entero.

- *Volumen_ecologico:* Contiene datos sobre el volumen de agua destinado al uso ecológico. Se mide en hm^3 y se trata de una variable numérica de tipo float.

- *Agua_sobrante:* Agua que sobra y no se puede quedar en el sistema de presas por seguridad. Se mide en hm^3 y es una variable numércia de tipo float.

- *Total_volumen_abastecido:* Total de los volúmenes destinados para los distintos usos comentados anteriormente. Se mide en hm^3 y es una variable numércia de tipo float.

- *Aportacion_guadalhorce_tajo:* Variable vacía, sin datos.

- *Total_aportacion:* Variable vacía, sin datos.

3.3 Limpieza, preprocesado y preparación de los datos

El objetivo de esta etapa es preparar los datos para su análisis posterior, de forma que puedan ser usados directamente por cualquier investigador o *data scientist*, sin tener que preocuparse por aspectos relacionados con su calidad, procedencia, distribución, etc [43].

Esta preparación de los datos implica distintas operaciones sobre los datos en bruto, entre las que encontramos, integración, limpieza, transformación y reducción. A continuación, se analizará cada una de estas operaciones sobre el conjunto de datos de interés.

3.3.1 Integración de los datos

La integración de datos implica la combinación de datos almacenados en distintas fuentes. Los datos almacenados en fuentes dispares se extraen utilizando diversas tecnologías para presentarlos en una vista unificada. Se entiende por integración de datos, todo lo que tiene que ver con el procesamiento de los datos con el fin de alinearlos, combinarlos y consolidarlos. Cuando se comparan atributos de una base de datos con otra durante la integración, se debe prestar especial atención a la estructura de los datos. Esto es para asegurar que el esquema del sistema de origen coincida con el esquema del sistema de destino [47].

Como se comentó en el apartado anterior, los datos proporcionados por la Junta de Andalucía vienen distribuidos en 96 hojas Excel (cuyo aspecto se puede ver en la figura 5) y 8 ficheros Excel. Es decir, cada uno de los 8 ficheros Excel contiene 12 hojas Excel.

Por lo tanto, nos encontramos antes varios problemas de integración:

1. Dado el formato del estadillo, cuyo fin es ser impreso físicamente para su presentación en formato de informe, contiene multitud de cabeceras y celdas *merged*, lo que imposibilita su lectura en Python.

2. El esquema de todos los ficheros no es exactamente igual.

3. Distribución de los datos en 96 hojas Excel distribuidas en 8 ficheros.

A continuación, se describirán las estrategias y métodos usados para solventar cada uno de los problemas de integración mencionados anteriormente.

1. Este problema se solventa eliminando manualmente todas las cabeceras, estilos y márgenes de los ficheros, dejando únicamente los datos en crudo. A continuación, se introduce manualmente la cabecera de cada variable, de forma que las variables quedan según se describió en el apartado 3.2. Esto se hace para cada hoja Excel dentro de cada fichero.

2. Los ficheros originales “ESTADILLO_DE_EMBALSES_2011.xlsx”, “ESTADILLO_DE_EMBALSES_2012.xlsx” y “ESTADILLO_DE_EMBALSES_2013.xlsx” contienen dos variables que no están en el restos de los ficheros *Salinidad_conde* y *Toneladas_sal_conde*. Por lo tanto, se introduce manualmente estas dos variables en el resto de ficheros, con un valor 0, para que el esquema de todos los datos coincida. Adicionalmente, se crea en todos los ficheros dos variables *Mes* y *Anyo*, con el objetivo de hacer análisis temporales, como se verá en apartados posteriores.

3. Este último problema se resuelve en varias etapas. En primer lugar, se consolidan, manualmente, todas las hojas correspondientes a un año en un mismo fichero Excel. Tras hacer esto, nos quedarán 8 ficheros Excel,

con una hoja cada uno. En según lugar, se pasa este fichero Excel a formato .csv. Por último, estos ficheros .csv serán leídos por el script recogido en el anexo 1 de forma que se creará un único dataframe con 74 variables que contendrá todos los datos relativos a la explotación del sistema de presas desde 2012 a 2018, como se puede ver en la siguiente figura.

```
In [12]: df_data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2894 entries, 0 to 2893
Data columns (total 74 columns):
 #   Column                                                Non-Null Count  Dtype
---  ------                                                --------------  -----
 0   Fecha                                                 2894 non-null   object
 1   Dia                                                   2894 non-null   int64
 2   Hora                                                  2894 non-null   object
 3   Mes                                                   2894 non-null   int64
 4   Anyo                                                  2894 non-null   int64
 5   Lluvia                                                569 non-null    float64
 6   Temperatura_Max                                       2894 non-null   float64
 7   Temperatura_Min                                       2894 non-null   float64
 8   Evaporacion                                           2894 non-null   float64
 9   Cota_guadalhorce                                      2894 non-null   float64
 10  Volumen_embalsado_guadalhorce                         2894 non-null   float64
 11  Variacion_volumen_guadalhorce                         2894 non-null   float64
 12  Tiempo_desembalse_guadalhorce                         1833 non-null   object
 13  Caudal_desembalse_guadalhorce                         1754 non-null   object
 14  Filtracion_guadalhorce                                2894 non-null   float64
 15  Volumen_desembalse_guadalhorce                        2894 non-null   float64
 16  Salinidad_guadalhorce                                 1189 non-null   float64
 17  Toneladas_sal_guadalhorce                             2724 non-null   object
 18  Volumen_evaporado_guadalhorce                         2893 non-null   object
 19  Aportacion_aparente_guadalhorce                       2894 non-null   float64
 20  Cota_guadalteba                                       2894 non-null   float64
 21  Volumen_embalsado_guadalteba                          2894 non-null   float64
 22  Variacion_volumen_guadalteba                          2894 non-null   float64
 23  Tiempo_desembalse_guadalteba                          1059 non-null   object
 24  Caudal_desembalse_guadalteba                          1011 non-null   object
 25  Filtraciones_guadalhorce                              0 non-null      float64
 26  Volumen_desembalse_guadalteba                         2894 non-null   float64
 27  Salinidad_guadalteba                                  206 non-null    float64
 28  Toneladas_sal_guadalteba                              2723 non-null   float64
 29  Volumen_evaporado_guadalteba                          2894 non-null   float64
 30  Aportacion_aparente_guadalteba                        2894 non-null   float64
 31  Volumen_total_guadalhocer_guadalteba                  2894 non-null   float64
 32  Variacin_total_guadalhocer_guadalteba                2894 non-null   float64
 33  Desembalse_total_guadalhocer_guadalteba               2894 non-null   float64
 34  Aportacin_aparente_total_guadalhocer_guadalteba       2894 non-null   float64
 35  Cota_conde                                            2894 non-null   float64
 36  Volumen_embalsado_conde                               2894 non-null   float64
 37  Variacion_volumen_conde                               2894 non-null   float64
 38  Filtraciones_conde                                    2894 non-null   float64
 39  Volumen_desembalse_conde                              2894 non-null   float64
 40  Salinidad_conde                                       366 non-null    float64
 41  Toneladas_sal_conde                                   366 non-null    float64
 42  Volumen_evaporado_conde                               2894 non-null   float64
 43  Aportacion_aparente_conde                             2894 non-null   float64
 44  Volumen_total_guadalhocer_guadalteba_conde            2894 non-null   float64
 45  Variacin_total_guadalhocer_guadalteba_conde          2894 non-null   float64
 46  Desembalse_total_guadalhocer_guadalteba_conde         2894 non-null   float64
 47  Aportacin_aparente_total_guadalhocer_guadalteba_conde 2894 non-null   float64
 48  Caudal_total_demandas                                 2847 non-null   float64
 49  Volumen_total_demandas                                2826 non-null   float64
 50  Salinidad_mezcla                                      1174 non-null   float64
 51  Toneladas_sal_mezcla                                  2579 non-null   float64
 52  Cota_Gaitanejo                                        1926 non-null   float64
 53  Volumen_gaitanejo                                     1926 non-null   float64
 54  Volumen_central-reversible_tajo                       1925 non-null   object
 55  Volumen_total_gaitanejo_tajo                          1933 non-null   float64
 56  Variacion_total_gaitanejo_tajo                        1909 non-null   object
 57  Caudal_abastecimiento                                 2866 non-null   object
 58  Horas_abastecimiento                                  2894 non-null   int64
 59  Min_abastecimiento                                    2894 non-null   int64
 60  Volumen_abasteciemiento                               2866 non-null   float64
 61  Caudal_riego                                          2804 non-null   float64
 62  Horas_riego                                           2894 non-null   object
 63  Min_riego                                             2894 non-null   int64
 64  Volumen_riego                                         2804 non-null   float64
 65  Caudal_ecologico                                      2866 non-null   float64
 66  Horas_ecologico                                       2894 non-null   int64
 67  Min_ecologico                                         2894 non-null   int64
 68  Volumen_ecologico                                     2866 non-null   float64
 69  Agua_sobrante                                         2874 non-null   object
 70  Total_volumen_abastecido                              2851 non-null   float64
 71  Aportacion_guadalhorce_tajo                           3 non-null      object
 72  Aportacion_arroyos_tajo                               0 non-null      float64
 73  Total_aportacion                                      0 non-null      float64
dtypes: float64(52), int64(8), object(14)
memory usage: 1.6+ MB
```

Figura 7: Dataframe sistema de presas valle del guadalhorce consolidado

3.3.2 Limpieza de los datos

Los datos del mundo real tienden a ser incompletos, contener ruido e inconsistencias. Las rutinas de limpieza de datos (*data cleansing*, en inglés) intentan completar los valores faltantes o *missing values*, suavizar el ruido al identificar valores atípicos o *outliers* y corregir inconsistencias en los datos [47].

Por lo tanto, será necesario someter al conjunto de datos, descrito en la figura 7, a un proceso de limpieza, con el fin de poder ser analizado posteriormente.

Una de las problemáticas que se detecta al inspeccionar el dataset es que el conjunto de datos proporcionado por la Junta de Andalucía contiene como separador decimal tanto el carácter “.” como el carácter “,”. Esto puede crear una incoherencia en el análisis de los datos, por lo que conviene unificar el criterio de separación decimal. Por lo tanto, como se puede ver en el anexo 1, se crea un bucle *for* para recorrer cada variable en busca del carácter “,” con el fin de sustituirlo por “.”.

Siguiendo con la inspección de los datos se observan ciertos valores atípicos derivados de la toma manual de los datos. En este caso se observan caracteres como “V”, “VAR”, “var”,”0AR”, “'#ÁREF!'”,”-” y “;” que se deberán eliminar. Por lo tanto, se crea un bucle *for* para recorrer cada variable, en busca de estos valores y sustituirlos por un 0. Esta imputación del valor 0 por los caracteres comentados es una recomendación del personal de la oficina técnica de la Junta de Andalucía.

Una de las consecuencias de que el conjunto de datos contenga, tanto separadores decimales heterogéneos, como caracteres atípicos, es que las variables afectadas se leen como objeto, como se puede ver en la figura 7.

Por último, se analizan los *missing values* del conjunto de datos y se estudian distintas estrategias de imputación de valores sobre la variable etiquetada. Esto se analiza a continuación, en el apartado 3.3.2.1.

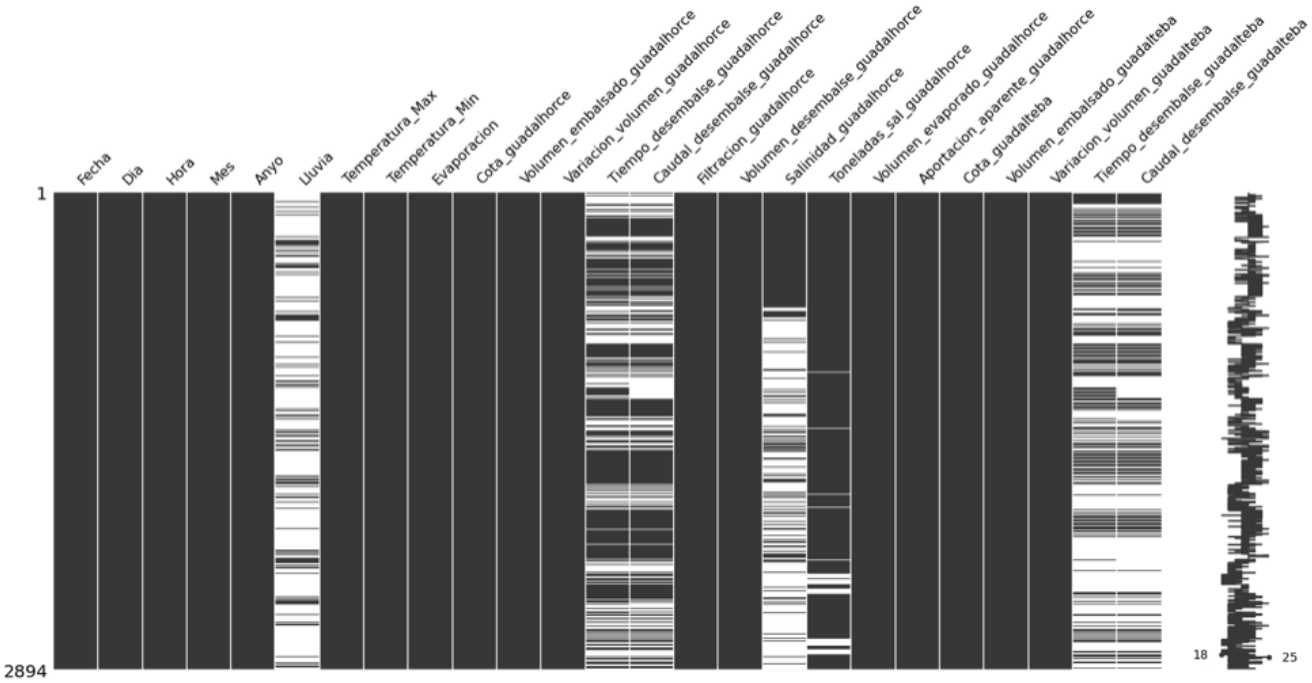
La mayoría de las técnicas de aprendizaje automático no pueden ejecutarse si existen *missing values*. Por lo tanto, este problema debe abordarse antes del modelado [48].

3.3.2.1 Gestión de *missing values* en variable etiquetada

En la figura 8 se pueden ver todos los *missing values* en las distintas variables del conjunto de datos. Esta representación se ha realizado usando la librería Missingno [60].

En la exploración de los *missing values* de las distintas variables es posible ver como en algún caso es algo normal. Es el caso de la variable *Lluvia* donde la falta de datos se debe a que no todos los días se producen precipitaciones, de ahí la gran cantidad de *missing values.* Otro caso sería el de las variables *Tiempo_desembalse_guadalhorce, Caudal_desembalse_guadalhorce*, *Tiempo_desembalse_guadalteba* y *Caudal_desembalse_guadalhorce* ya que no todos los días se realizan desembalses de estas presas.

Por otro lado, se puede ver como existen variables completamente vacías, como *Aportacion_guadalhorece_tajo*, *Aportacion_arroyos_tajo* y *Total_aportacion*. Según el personal de la Junta de Andalucía, estas variables ya no se recogen, por lo que serán susceptibles de ser eliminadas del dataset.



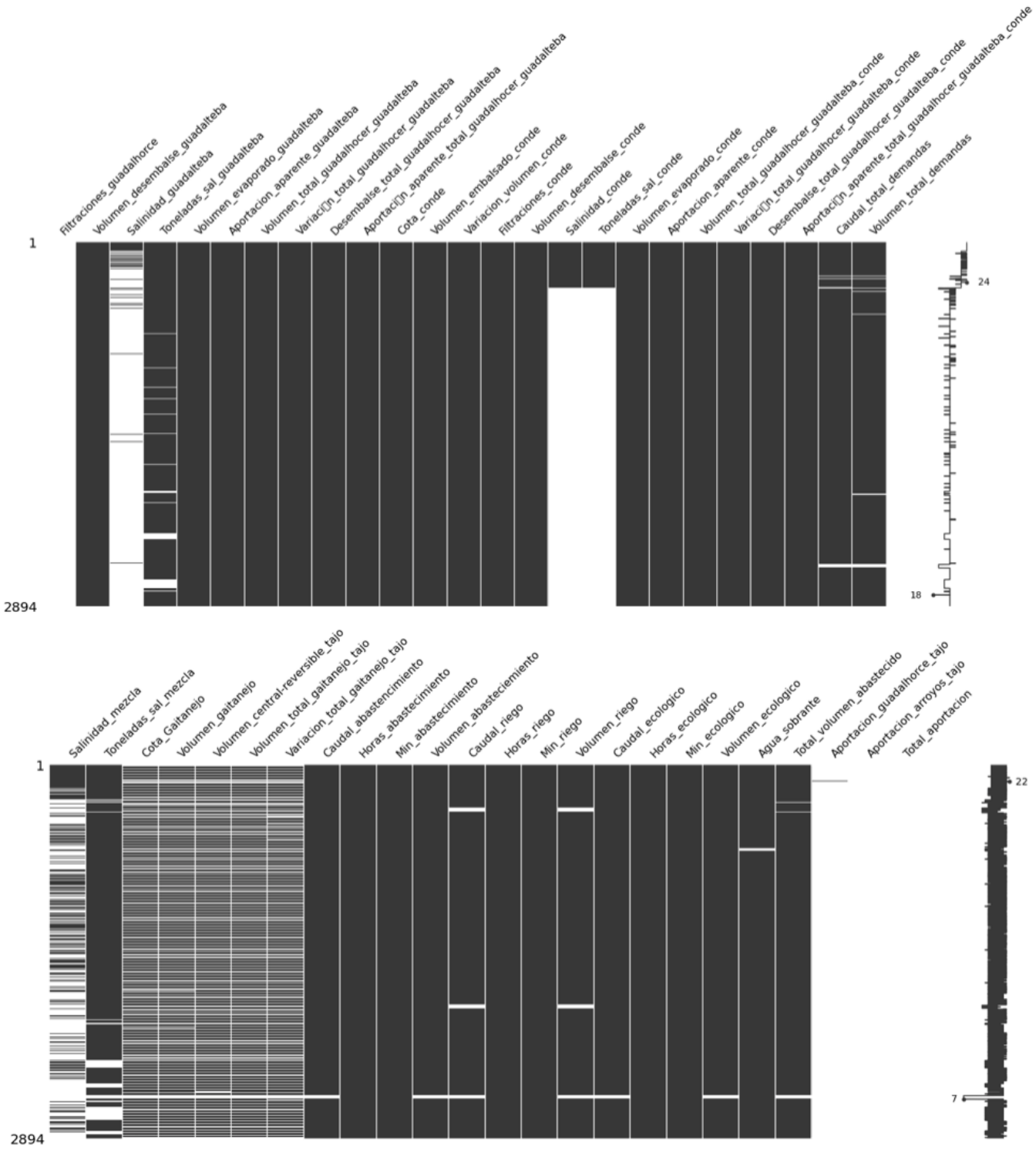


Figura 8: *Missing values* presentes en el dataset

Tras distintas reuniones con el personal técnico de la Junta de Andalucía les preocupa la falta de valores en la variable *Salinidad_mezcla*, ya que debería ser un valor continuo, sin embargo, debido a una toma de datos manual, no siempre se toma este dato.

Como se puede ver en la figura 9, el porcentaje de *missing values* presente en la variable *Salinidad_mezcla* es del 59.43%. Como se dijo anteriormente, esta

falta de datos no es algo natural, si no, que se debe a un error en la toma de datos y el valor de esta variable se antoja indispensable para controlar la calidad del agua usada para regadío, consumo y vertidos al cauce del río.

```
In [29]: #Porcentaje de NAN trás la imputación
         salinidad_mezcla_percentage = 100*df_data['Salinidad_mezcla'].isnull().sum()/len(df_data['Salinidad_mezcla'])
         salinidad_mezcla_percentage

Out[29]: 59.43331029716655
```

Figura 9: Porcentaje de *missing values* en *Salinidad_mezcla*

La gestión de estos *missing values* se puede abordar con distintas estrategias. Como se verá en apartados posteriores, en este documento se evaluará la precisión de los distintos clasificadores en función de la estrategia de gestión de los *missing values* usadas. Las estrategias propuestas se presentan a continuación.

1. **Eliminar las filas** donde *Salinidad_mezcla* contiene un *missing values*. Esto significa eliminar 1720 filas del dataset, por lo que, no solo se estará perdiendo información acerca de la variable *Salinidad_mezcla*, si no, del resto de variables. Este método es el más simple, pero eliminar filas con *missing values* puede ser demasiado limitante en algunos problemas de modelado predictivo

   En este caso bastaría con recurrir a la función dropna() de Pandas sobre la variable *Salinidad_mezcla*, como se puede ver en el anexo 1.

2. **Uso de la media para la imputación *missing values.*** Con este método se completarían los 1720 *missing values* con la media del atributo *Salinidad_mezcla*, cuyo valor es 913.96.

   Esto se realizaría usando la función mean() de la librería Pandas. Como se puede ver en la figura 10, al realizar la imputación de *missing values* con la función mean(), los valores estadísticos principales (excepto la media) cambian respecto a la variable sin preprocesar. Por lo que se puede afirmar que tras esta imputación, las variables no conservan la misma distribución, como se muestra en la figura 11.

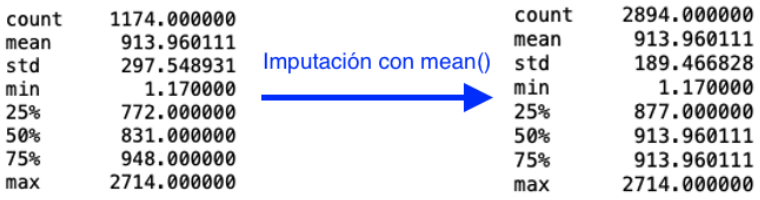


Figura 10: Imputación *missing values* en variable *Salinidad_mezcla* con función mean()

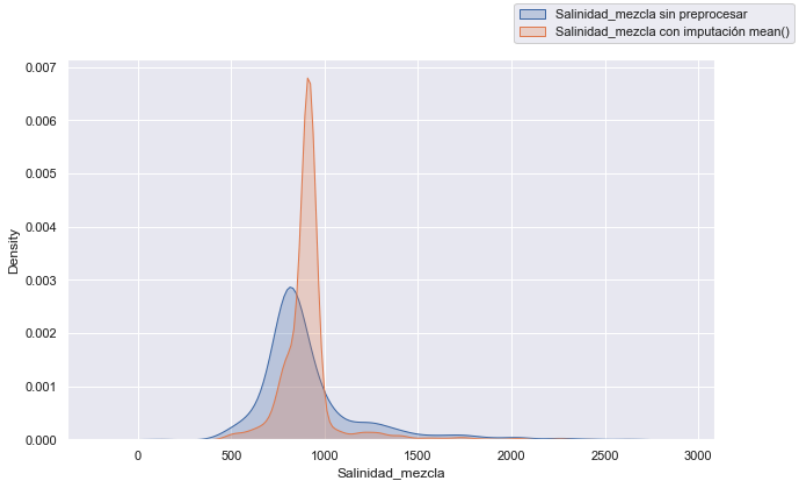


Figura 11: Distribución de *Salinidad_mezcla* con imputación con función mean()

3. **Uso de la mediana para la imputación *missing values***. Con este método se completarían los 1720 *missing values* con la media del atributo *Salinidad_mezcla*, cuyo valor es 831.00.

   Esto se realizaría usando la función median() de la librería Pandas. Como se puede observar en la figura 13, usando este estadístico se conserva la posición central del conjunto de datos tras la imputación, incluso la diferencia en la desviación típica respecto a la imputación con mean() es menor, como se puede ver en la figura 12.

Figura 12: Imputación *missing values* en variable *Salinidad_mezcla* con función median()

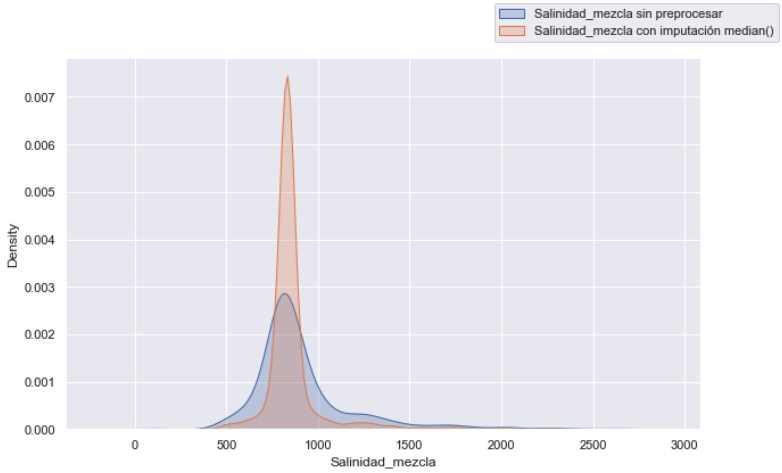


Figura 13: Distribución de *Salinidad_mezcla* con imputación con función median()

4. **Uso de rolling mean para la imputación *missing values***. En este caso no se usará la media del conjunto completo para imputar los *missing values,* si no que se usarán ventanas de 30 registros para imputar estos *missing values.* Si algún *missing value* cae en esta ventana de tamaño 30, dicho *missing value* será reemplazado por la media en esta ventana. Se usa una ventana de 30 registros, ya que cada registro se corresponde a un día, por lo tanto, estaremos usando ventanas de 30 días.

   Para realizar esta imputación se recurre a la función rolling() de la librería Pandas. Como se puede ver en las desviaciones típicas de la figura 14 y en la gráfica de distribución de la figura 15, en este caso, la distribución de la variable sin preprocesar y la variable con el uso de rolling mean para realizar la imputación, se asemejan más que en los dos casos anteriores.

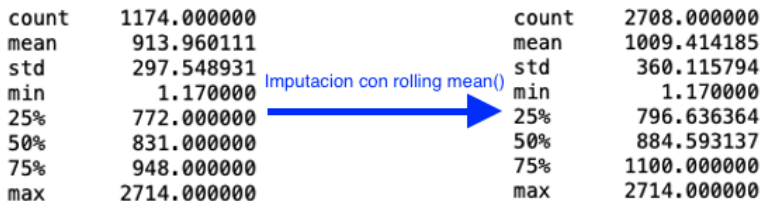


Figura 14: Imputación *missing values* en variable *Salinidad_mezcla* con función rolling mean()

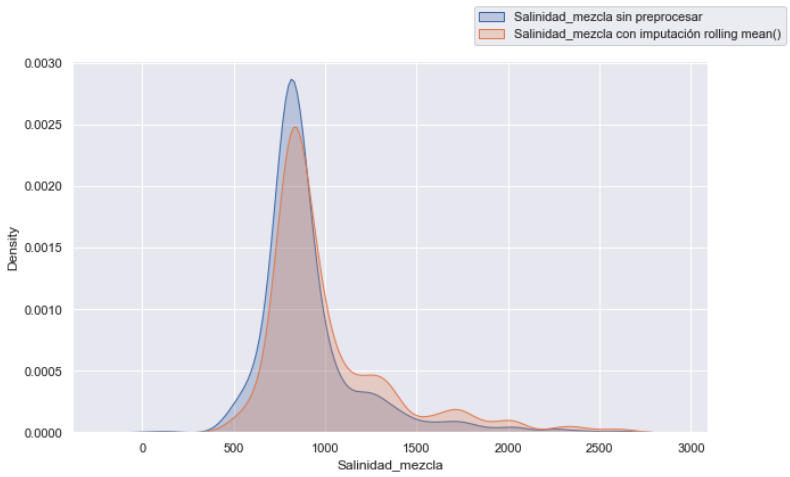


Figura 15: Distribución de *Salinidad_mezcla* con imputación con función rolling mean()

5. **Uso de rolling median para la imputación *missing values***. Este caso es similar al anterior, pero en este caso se calcula la mediana de la ventana de 30 registros para imputar los *missing values* contenidos en dicha ventana.

La distribución usando rolling median no varia significativamente a la que hayamos usando rolling mean, por lo que habrá que analizar con que estrategia de imputación se consigue mayor precisión en los métodos de predicción. Esto se verá en apartados posteriores.

count	1174.000000	Imputación con rolling median()	count	2708.000000
mean	913.960111		mean	999.441348
std	297.548931		std	363.155816
min	1.170000		min	1.170000
25%	772.000000		25%	796.000000
50%	831.000000		50%	876.000000
75%	948.000000		75%	1065.000000
max	2714.000000		max	2714.000000

Figura 16: Imputación *missing values* en variable *Salinidad_mezcla* con función rolling median()

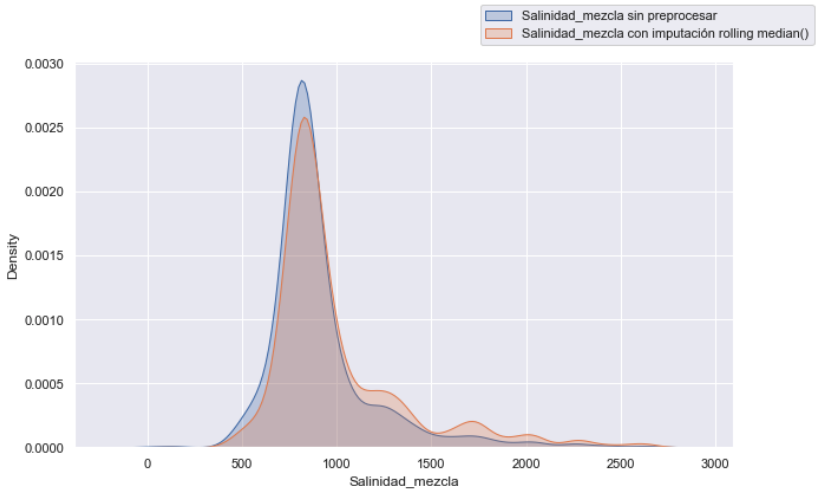


Figura 17: Distribución de *Salinidad_mezcla* con imputación con función rolling median()

6. **Uso de interpolación lineal para la imputación *missing values***. La interpolación lineal se trata de un método de ajuste que utiliza polinomios lineales para construir nuevos puntos de datos dentro del rango de un conjunto discreto de puntos de datos conocidos [49]. Por lo tanto, en este estudio, ajustaremos los *missing values* a través de polinomios lineales cuyos coeficientes serán dados por los datos que ya conocemos.

   En este caso se puede observar que la desviación típica, respecto al conjunto original, aumenta algo mas que en los dos casos anteriores. Se deberán combinar estos resultados con la precisión de los métodos predictivos, que se analizarán en apartados posteriores, para tomar una decisión acerca del mejor método de imputación.

count	1174.000000		count	2894.000000
mean	913.960111		mean	1030.223290
std	297.548931	Imputación con Interpolación lineal	std	379.230801
min	1.170000		min	1.170000
25%	772.000000		25%	800.400000
50%	831.000000		50%	901.000000
75%	948.000000		75%	1140.500000
max	2714.000000		max	2714.000000

Figura 18: Imputación *missing values* en variable *Salinidad_mezcla* con interpolación lineal

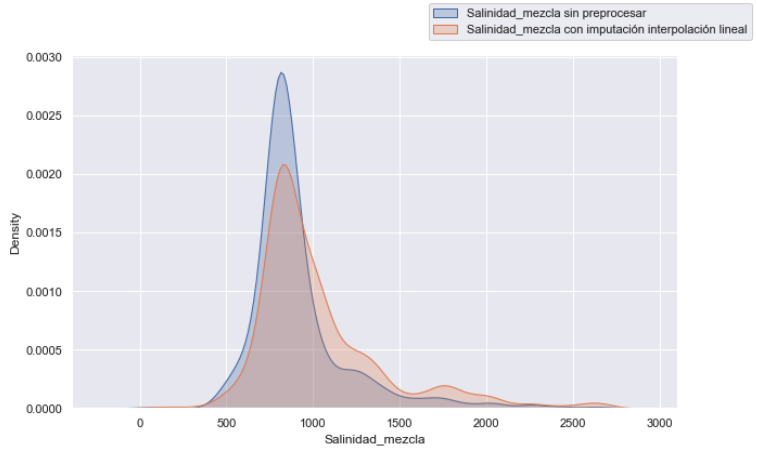


Figura 19: Distribución de *Salinidad_mezcla* con imputación con interpolación lineal

7. **Uso de interpolación cuadrática para la imputación *missing values***. La interpolación cuadrática se trata de un método de ajuste que utiliza polinomios de grado dos para construir nuevos puntos de datos dentro del rango de un conjunto discreto de puntos de datos conocidos [50].

   En este caso, los valores estadísticos que se obtienen con la imputación con interpolación cuadrática difieren ampliamente del conjunto original. A la espera de ver los resultados de precisión de los distintos métodos predictivos, se puede afirmar que la imputación con interpolación cuadrática no es una buena estrategia para gestionar los *missing values* del conjunto de datos presentado en este documento.

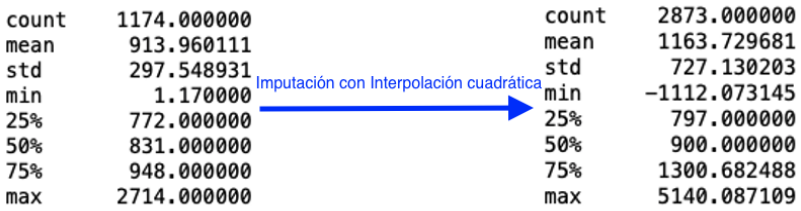


Figura 20: Imputación *missing values* en variable *Salinidad_mezcla* con interpolación cuadrática

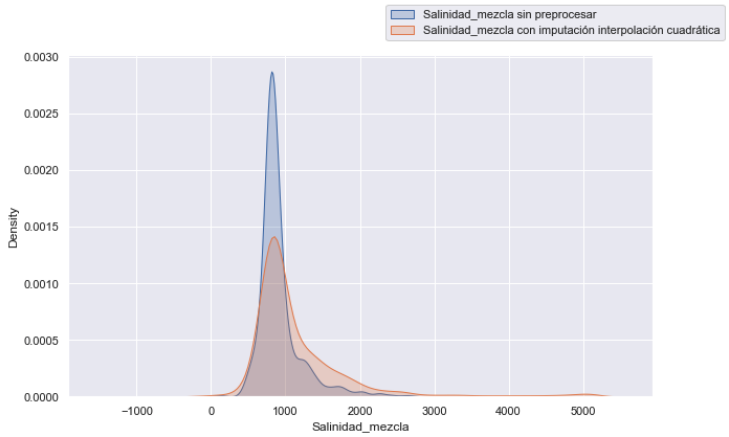


Figura 21: Distribución de *Salinidad_mezcla* con imputación con interpolación cuadrática

8. **Uso de interpolación cúbica para la imputación *missing values***. La interpolación cúbica se trata de un método de ajuste que utiliza polinomios de grado tres para construir nuevos puntos de datos dentro del rango de un conjunto discreto de puntos de datos conocidos [50].

   Al igual que en el caso interior, como se puede ver en la figura 22, este no sería un buen método de imputación de *missing values*, por la diferencia en la distribución respecto al conjunto original.

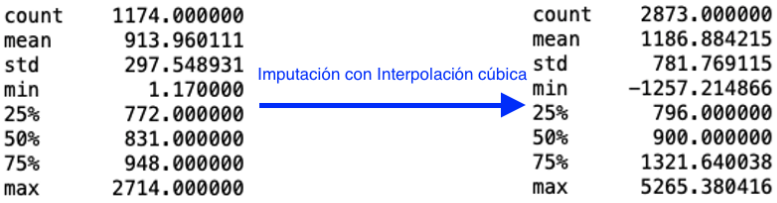


Figura 22: Imputación *missing values* en variable *Salinidad_mezcla* con interpolación cúbica

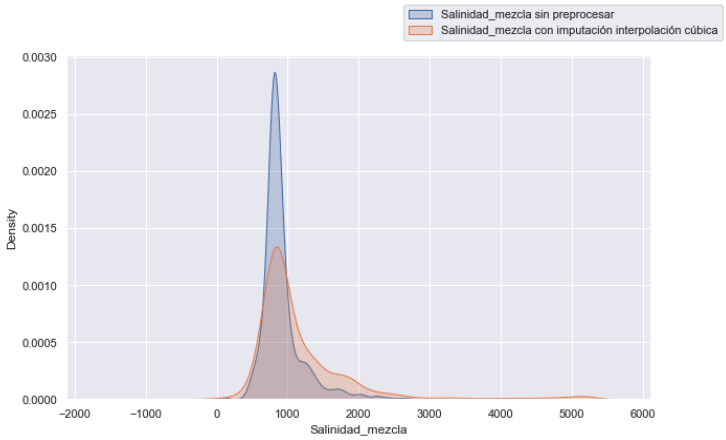


Figura 23: Distribución de *Salinidad_mezcla* con imputación con interpolación cúbica

9. **Uso de interpolación spline para la imputación *missing values***. La interpolación spline es una forma de interpolación en la que el interpolante es un tipo especial de polinomio por partes llamado spline. La interpolación spline se prefiere a menudo a la interpolación polinomial porque el error de interpolación puede reducirse incluso cuando se utilizan polinomios de bajo grado para la spline [51].

   Al igual que en los dos casos anteriores, se puede afirmar que la interpolación spline no es una buena estrategia para gestionar los *missing values* de nuestro conjunto de datos, dada la diferencia estadística mostrada en las figuras 24 y 25.

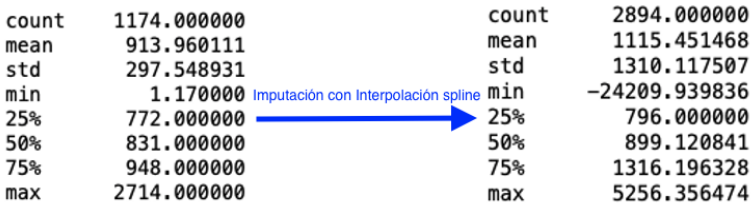


Figura 24: Imputación *missing values* en variable *Salinidad_mezcla* con interpolación spline

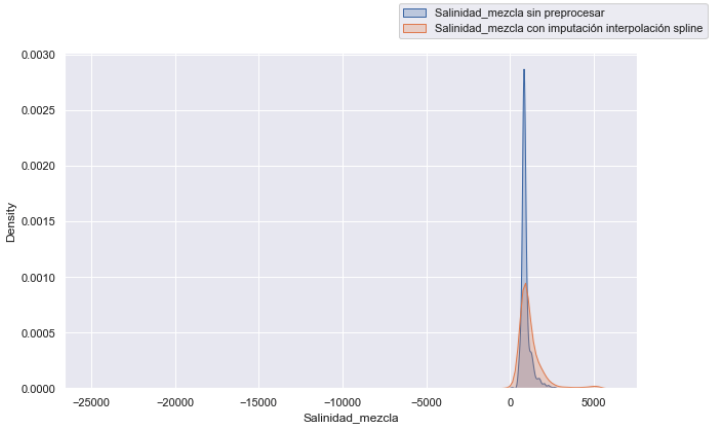


Figura 25: Distribución de *Salinidad_mezcla* con imputación con interpolación spiline

10. **Uso de interpolación akima para la imputación *missing values***. La interpolación de Akima es una interpolación sub-spline continuamente diferenciable. Se construye a partir de polinomios de tercer orden por partes. Sólo se utilizan datos de los siguientes puntos vecinos para determinar los coeficientes del polinomio de interpolación. No hay necesidad de resolver grandes sistemas de ecuación y, por lo tanto, este método de interpolación es computacionalmente muy eficiente [52].

    El conjunto de datos resultante de usar la interpolación Akima para la imputación de los *missing values* sigue una distribución similar al conjunto original de datos, como se puede ver en las figuras 26 y 27. Queda ver la precisión obtenida por los distintos métodos de aprendizaje

automático con cada estrategia de imputación para definir cual es la mejor estrategia de imputación.

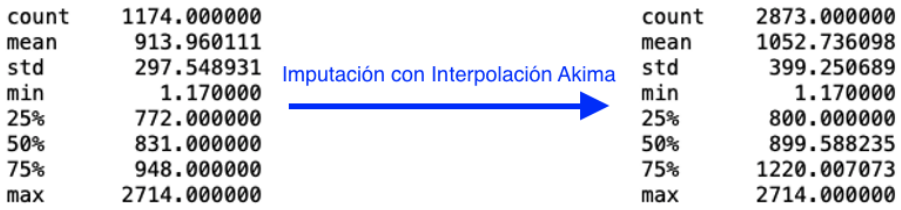


Figura 26: Imputación *missing values* en variable *Salinidad_mezcla* con interpolación Akima

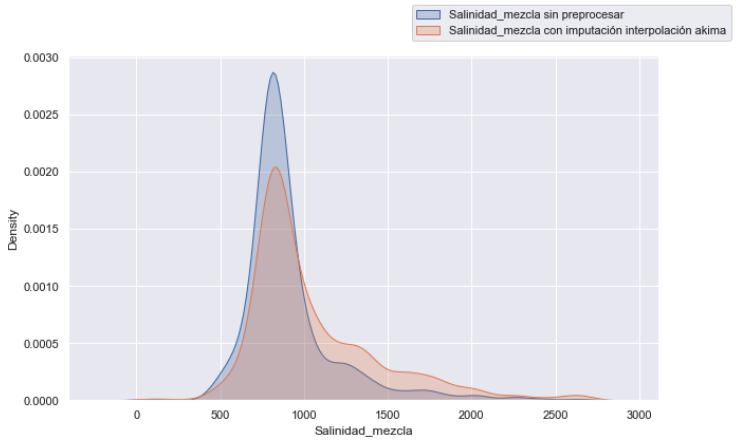


Figura 27: Distribución de *Salinidad_mezcla* con imputación con interpolación Akima

11. **Uso de Regresión Lineal para la predicción de *missing values***. Dado que la variable *Salinidad_mezcla* es numérica de tipo float con una pendiente determinada, se usará regresión lineal para la predicción de *missing values*. La regresión lineal es un método estadístico que trata de modelar la relación entre una variable numérica continua y una o más variables independientes mediante el ajuste de una ecuación lineal [53].

    Se usará como variable etiquetada los valores de *Salinidad_mezcla* que no contienen *missing values* y como variables regresoras *Cota_guadalhorce*, *Cota_guadalteba*, *Volumen_total_guadalhorce_guadalteba_conde*, Desembalse*_total_guadalhocer_guadalteba_conde*, ya que son variables que están completas y son variables del interés de la Junta de Andalucía.

    Vemos en las figuras 28 y 29 que este método de gestión de *missing values* es el que menor diferencia de desviación típica presenta, por lo que, a priori, puede ser una buena estrategia de gestión de *missing values.* En el apartado de análisis ser verá la capacidad predictiva de los modelos de aprendizaje automático usando este método de gestión de *missing values.*

Figura 28: Predicción de *missing values* en variable *Salinidad_mezcla* mediante Regresion Lineal

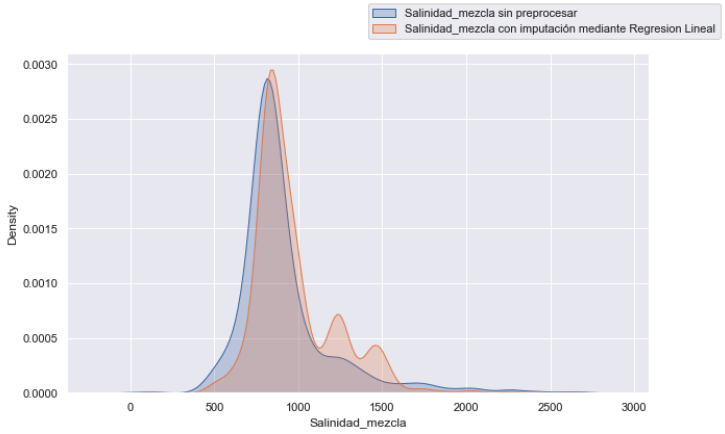


Figura 29: Distribución de *Salinidad_mezcla* con Regresion Lineal

3.3.3 Transformación de los datos

En este paso del preprocesamiento, los datos se transforman o consolidan para que los métodos de aprendizaje automático sean más eficientes y los patrones encontrados sean más fáciles de entender [47].

Relativo al conjunto de datos presentado en este documento, y una vez se ha realizado la gestión de los *missing values,* se van a realizar dos operaciones de transformación, con el objetivo de facilitar el análisis de los datos. Estas operaciones son la discretización de la variable etiquetada y la normalización de variables.

3.3.3.1 Discretización de la variable etiquetada

La discretización de datos es el proceso de establecer varios puntos de corte para atributos con valores numéricos continuos con el fin de obtener valores enteros o discretos de dichos atributos [55].

La discretización es una tarea esencial del preprocesamiento de datos, no solo porque algunos métodos de aprendizaje no manejan atributos continuos en la variable etiquetada, sino porque los datos transformados en un conjunto de intervalos son cognitivamente más relevantes para la interpretación de un ser humano y la carga computacional se reduce a usar variables discretas [56].

Existen multitud de estrategias de discretización, una de ellas y la que se ha usado en este estudio, consiste en realizar una discretización con un número fijo de intervalos. En esta situación, se debe elegir a priori a priori el número apropiado de intervalos: demasiados intervalos no serán adecuados para el problema de aprendizaje y muy pocos intervalos pueden correr el riesgo de perder información interesante.

En este caso, se ha contado con la Junta de Andalucía para realizar la asignación de intervalos:

- Mezcla de aguas no salina: 0 ≤ *Salinidad_mezcla* ≤ 900 → *Salinidad_mezcla* = 0

- Mezcla de aguas salina: *Salinidad_mezcla* > 900 → *Salinidad_mezcl*a = 1

Para realizar la discretización descrita anteriormente se ha recurrido a la función cut() de la librería Pandas [57].

3.3.3.2 Normalización del conjunto de datos

La normalización de los datos intenta dar a todos los atributos o variables el mismo peso y es particularmente útil para algoritmos de clasificación que involucran redes neuronales o mediciones de distancia.

Si se utiliza el algoritmo de *backpropagation* de la red neuronal como método de clasificación, como se verá en apartados posteriores, normalizar los valores de entrada para cada atributo o variable del conjunto de entrenamiento ayudará a acelerar la fase de aprendizaje [47].

Para los métodos basados en distancia, las variables que se miden a diferentes escalas no contribuyen por igual a la función de ajuste y aprendizaje del modelo, por lo que podrían terminar creando un sesgo, por lo tanto, la normalización permite atribuir a todas las variables el mismo peso. También es útil cuando no se tiene conocimiento previo de los datos [53].

En este caso, se recurre a la normalización *min-max* ya que se conserva la distribución y relaciones de los valores del conjunto original [47]. De esta forma, todas las variables se transformarán en el rango [0,1], lo que significa que el valor mínimo y máximo de cada variable será 0 y 1, respectivamente [53].

A nivel de código, se recurrirá a la función MinMaxScaler() de la librería Scikit-learn [54]. Por lo tanto, para cada registro de cada variable tendremos la expresión definida en la ecuación 1.

$$x_{scaled} = \frac{x - \min(x)}{\max(x) - \min(x)}$$

Ecuación 1: Expresión matemática de normalización *min-max*

3.3.4 Reducción de datos

Las técnicas de reducción de datos se aplican para obtener una representación reducida del conjunto de datos que mantenga la integridad de los datos originales. Es decir, el procesamiento y análisis de datos en el conjunto de datos reducido debería ser más eficiente, pero producir los mismos o mejores resultados analíticos [47].

Una de estas técnicas de reducción de datos es la selección de subconjuntos de datos, que reduce el tamaño del conjunto de datos al eliminar atributos (o dimensiones) irrelevantes o redundantes. El objetivo de la selección de subconjuntos de atributos es encontrar un conjunto mínimo de atributos de modo que la distribución de probabilidad resultante de las clases de datos sea lo más cercana posible a la distribución original obtenida utilizando todos los atributos. El análisis de datos en un número reducido de atributos tiene un beneficio adicional: reduce el número de atributos que aparecen en los patrones descubiertos, ayudando a que los patrones sean más fáciles de entender [58].

En el estudio presentado en este documento se propone un método de selección de variables univariado (*Univariate Feature Selection*, según su termino en inglés). Este método funciona seleccionando las mejores variables basadas en pruebas estadísticas univariadas. Comparamos cada variable del dataset con la variable objetivo o etiquetada, en este caso *Salinidad_mezcla*, para ver si existe alguna relación estadísticamente significativa entre ellas. A esto también se conoce como análisis de varianza (ANOVA, por su término en inglés). Cuando se analiza la relación entre una característica y la variable objetivo, se ignoran las otras características. Por eso se llama un análisis univariado. Teniendo en cuenta dichas pruebas estadísticas, a cada variable se le asigna una puntuación. Finalmente, se comparan todas las puntuaciones de las pruebas estadísticas y se seleccionarán las variables con las puntuaciones más altas [59].

Para realizar este análisis univariado a nivel software, la libreria scikit-learn proporciona la función SelectKBest(), mediante la cual, se puede utilizar un conjunto de diferentes pruebas estadísticas univariadas para seleccionar un número específico de variables del dataset. Entre otros métodos, se puede utilizar el test ANOVA F-value, que es apropiado para variables de entrada numéricas y variable objetivo categórica, como es el caso estudiado en este documento. Para aplicar ANOVA F-value, se recurre a la función f_classif, pasada como parámetro a la función SelectKBest(), presentada anteriormente [60].

Por lo tanto, se aplica SelectKBest() con la función f_classif para encontrar las 20 variables más puntuación y, por lo tanto, más influyentes, cuyo resultado se puede ver en la figura 30.

	Feature_Name	Score
50	Caudal_riego	1074.328809
52	Volumen_riego	1070.707001
23	Volumen_evaporado_guadalteba	975.531577
7	Cota_guadalhorce	678.817119
16	Cota_guadalteba	639.715798
8	Volumen_embalsado_guadalhorce	637.706848
13	Salinidad_guadalhorce	613.614388
25	Volumen_total_guadalhocer_guadalteba	605.560178
4	Temperatura_Max	566.625849
17	Volumen_embalsado_guadalteba	564.172203
38	Volumen_total_guadalhocer_guadalteba_conde	550.541325
6	Evaporacion	470.639271
5	Temperatura_Min	452.830817
43	Volumen_total_demandas	344.525832
42	Caudal_total_demandas	327.404162
56	Total_volumen_abastecido	293.319318
29	Cota_conde	254.991574
20	Volumen_desembalse_guadalteba	242.719479
30	Volumen_embalsado_conde	231.729005
2	Anyo	224.222623

Figura 30: Univariate Feature Selection con SelectKBest() y test ANOVA F-value

Notar que se seleccionan 20 variables, ya que, por petición de la Junta de Andalucía, se quería ver las 20 variables más influyentes en la variable *Salinidad_mezcla*.

Notar también que, dependiendo del método de gestión de *missing values* utilizado, se obtendrá un conjunto de 20 variables ligeramente diferente. Para hallar este conjunto de 20 variables se ha utilizado el método de gestión de *missing values* con el que más precisión muestran los modelos de aprendizaje automático, analizado en apartados posteriores.

3.4 Métodos de aprendizaje automático

El dataset resultante de la fase de preprocesado contendrá 20 variables, resultantes de un proceso de reducción de dimensionalidad y mostradas en la figura 30, y una variable etiquetada, que será *Salinidad_mezcla*.
Las 20 variables de entrada estarán normalizadas siguiendo el método *min-max*. La variable etiqueta tendrá el problema de *missing values* resuelto y estará discretizada en dos categorías, como se describió en el apartado 3.3.3.1.

Dado este conjunto de datos resultante de la fase de preprocesado, el siguiente paso que se pretende resolver, con la aplicación de los distintos métodos de aprendizaje automático, un problema de clasificación binaria de la variable *Salinidad_mezcla*. En este tipo de problemas se pretende encontrar un predictor o clasificador con el fin de predecir la clase de la variable etiquetada, en este caso, predecir la clase 0, correspondiente con muestras de agua no salina, o la clase 1, correspondiente con muestras de agua salina [47].

La clasificación de datos presentada en este documento es un proceso de tres pasos, que consiste en un paso de aprendizaje o entrenamiento (donde se construye un modelo de clasificación), un paso de validación (donde se evalúa el clasificador) y un paso de clasificación o test (donde el modelo se usa para predecir la clase de conjunto de datos nuevo).

Para el aprendizaje y validación se utilizará el conjunto de entrenamiento (también conocido como *training set*) y el conjunto de validación (también conocido como *validation set*), respectivamente, según se muestra en la figura 31. Dado que el predictor o clasificador se hallará usando los conjuntos de entrenamiento y validación, y conociendo la variable etiquetada, *Salinidad_mezcla*, se tratará de un problema de aprendizaje supervisado.

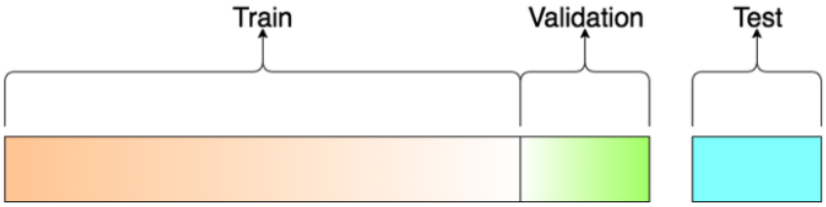


Figura 31: Division del dataset

Para la fase de clasificación o test, se utilizará el conjunto de test (también conocido como *test set*), que será un conjunto independiente a los de entrenamiento y validación. Aunque el conjunto de test consta de la variable etiquetada, no se usa para realizar la predicción. Posteriormente, se comparará la predicción con el valor de la variable etiquetada en el conjunto de test para hallar la precisión (*accuracy,* según su término en inglés) del clasificador [61].

Dado que el dataset tiene 2894 registros, lo cual no supone un número elevado, se recomienda que la división de los conjuntos de entrenamiento, validación y test sea, 70%, 15% y 15% respectivamente [62]. Por lo tanto, para obtener los conjuntos de train y test se recurrirá a la función train_test_split(). Se le pasan como parámetro el tamaño que se desea para el conjunto de test y la cifra 42 [64] para el parámetro random_state, de forma que la división entre los conjuntos de entrenamiento y test sea aleatoria, pero no cambie cada vez que se ejecute el código.

Dado que se utilizará el mecanismo de validación cruzada de 5 pliegues (5-*folds cross-validation* por su termino en inglés), como se recomienda en la documentación de la librería scikit-learnt [65], el conjunto de validación será 1/5 del conjunto de entrenamiento, como se muestra en la figura 32. Esto significa que el conjunto de validación supondrá un 17% del dataset. El proceso de validación cruzada es el siguiente:

1. Se divide el conjunto de datos (en este caso el 85% del dataset resultante de la división entre conjunto de test y de entrenamiento) en *k* pliegues o *folds* únicos, en este caso *k* es 5.
2. Luego se entrena el modelo usando los pliegues *k*-1 y se evalúa el modelo usando el pliegue *k* restante. Se anotan la precisión / error.
3. Se repite este proceso hasta que cada *k* pliegue sirva como conjunto de validación. Luego se toma el promedio de los valores de precisión y error registrados en cada iteración, que constituirán las métricas de rendimiento para el modelo o clasificador hallado.

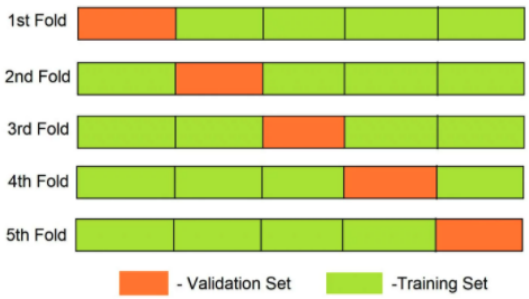


Figura 32: Validación cruzada de 5 pliegues o 5-folds cross-validation

El motivo de usar validación cruzada es evaluar el modelo con distintos conjuntos de evaluación durante la fase de entrenamiento. Al usar validación cruzada se da como resultado una estimación menos sesgada o menos optimista de la capacidad predictora del modelo [66].

Cada método de aprendizaje automático dispone de una serie de hiperparámetros. Dichos hiperparámetros varían según el modelo de aprendizaje automático elegido, como se verá en el apartado posterior. Encontrar la combinación óptima de hiperparámetros para un modelo hará que dicho modelo maximice su capacidad predictora.

Para encontrar la combinación óptima de hiperparámetros de cada modelo se utilizará el método GridSearch, a través de la función GridSearchCV() de la librería scikit-learnt. GridSearch es un método eficaz para ajustar los hiperparámetros en el aprendizaje supervisado y mejorar el rendimiento de generalización de un modelo. Con GridSearch se prueban todas las combinaciones posibles de los parámetros de interés y se encuentra la mejor combinación [67].

Notar que el uso conjunto de validación cruzada y GridSearch es una práctica habitual en los *pipelines* de proyectos de ciencia de datos, esto hace el modelo sea mas robusto y se obtenga menos sobreajuste (*overfitting,* por su término en inglés) [67]. De esta forma, cada hiperparámetro se entrenará con 5 conjuntos de entrenamiento y se evaluará con 5 conjuntos de evaluación diferentes.

A continuación, se presentarán los métodos de aprendizaje automático usados para la tarea de clasificación descrita anteriormente. Notar que se han seleccionado los métodos de aprendizaje automático más utilizados en los distintos estudios de la predicción de salinidad en presas, citados en el estudio del arte de este documento. Así, los estudios de A. El Bilali y Chou et *al*. recurren a redes neuronales, arboles de decisión, Random Forest, máquinas de soporte vectorial y AdaBoost para predecir la calidad del agua [35], [36]. Además de los estudios citados anteriormente, distintos estudios recurren a redes neuronales para predecir parámetros de salinidad de aguas de reservas o embalses, así como parámetros de calidad del agua [37], [40], [41], [42].

3.4.1 Árboles de decisión

Los árboles de decisión son un tipo de modelo no paramétrico que se puede utilizar tanto para clasificación como para regresión. Esto significa que los árboles de decisión son modelos flexibles que no aumentan su número de parámetros a medida que agregamos más variables, y pueden generar una predicción categórica o una predicción numérica.

Se construyen utilizando dos tipos de elementos: nodos y ramas. En cada nodo, se evalúa una de las variables del dataset para dividir las observaciones en el proceso de entrenamiento. La estructura de un árbol de decisión se puede ver en la figura 33, donde se pueden ver tres tipos de nodos [62]:

- Nodo raíz (*Root node*, en inglés): Es el nodo que inicia el gráfico. En un árbol de decisión normal evalúa la variable que mejor divide los datos.

- Nodo intermedio (*Intermediate node*, en inglés): Son nodos donde se evalúan las variables pero que no son los nodos finales donde se hacen las predicciones. En los nodos intermedios también dividen los datos según la condición establecida en una variable determinada.

- Nodos hoja (*Leaf o Leave node*, en inglés): Son los nodos finales del árbol, donde se realizan las predicciones de una categoría o un valor numérico.

Los árboles de decisión se construyen dividiendo de forma recursiva las muestras del conjunto de entrenamiento utilizando las características del dataset que funcionan mejor para realizar dicha división. Esto se hace evaluando ciertas métricas, como el índice de Gini o la Entropía para árboles de decisiones categóricas [62].

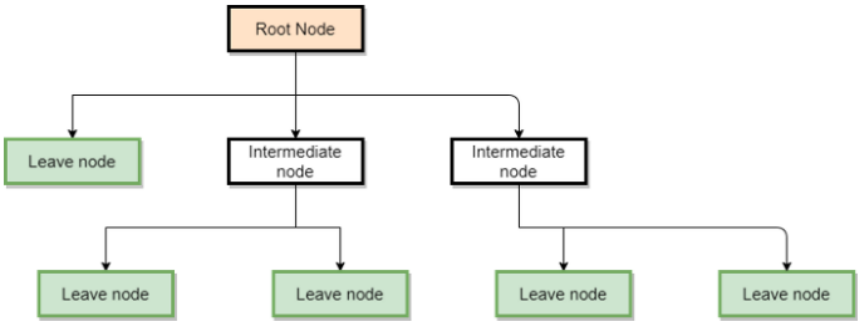


Figura 33: Estructura de los árboles de decisión

Para la construcción del modelo de arboles de decisión se recurre al método DecisionTreeClassifier() de la librería scikit-learnt [70], que se entrena con el conjunto de entrenamiento descrito anteriormente.

Con fin de optimizar el modelo, este método permite la configuración de distintos parámetros, como la longitud máxima del árbol (max_depth) o el

número mínimo de muestras para dividir un nodo (min_samples_split), ambos usados en este estudio. Ya que se desea optimizar el clasificador hallado, como se comentó en el apartado anterior, se aplica al método GridSearch al clasificador DecisionTreeClassifier() definiendo distintos valores de los parámetros max_depth y min_samples_split. Con tal de configurar un conjunto de valores amplio, para el parámetro max_depth se prueban valores desde 4 a 20 y para min_samples_split se prueban los valores 2, 10, 20, 50 y 100. Los valores óptimos del clasificador se hallan para árboles profundos y minimizando min_samples_split, aunque cuando se utilizan árboles muy profundos se corre el riesgo de *overfitting* o sobreajuste.

3.4.2 Random Forest

Random Forest se puede considerar como un conjunto de árboles de decisión. La idea detrás del aprendizaje en conjunto es combinar predictores débiles (*weak learners,* en inglés) para construir un modelo más robusto, es decir, un predictor fuerte (*strong learner*, en inglés), que tenga un mejor error de generalización y sea menos susceptible al sobreajuste u *overfitting*. El método Random Forest se puede resumir en cuatro pasos [67]:

1. Se utiliza un método conocido como *bagging* o agregación de *bootstrap* para crear una muestra aleatoria de los datos.

2. Se desarrolla un árbol de decisión de la muestra obtenida por el método *bagging.* En cada nodo de este árbol de decisión:

    2.1. Se seleccionan aleatoriamente *d* variables

    2.2. Se divide el nodo utilizando la función que proporciona la mejor división de acuerdo con la variable etiquetada.

3. Se repiten el paso 1 y paso 2 *k* veces, donde *k* es el número de árboles de decisión elegidos para Random Forest.

4. En este último paso se realiza votación por mayoría para predecir la clase de la variable etiquetada. Esto es que cada árbol tendrá asociado un voto. De esta forma, se propondrá como predicción final lo que voten la mayoría de los árboles de decisión.

Para trabajar con Random Forest se recurre al método RandomForestClassifier() de la librería scikit-learnt [71].

Aunque Random Forest no ofrece el mismo nivel de interpretabilidad que los árboles de decisión, una gran ventaja de Random Forest es que no hay que preocuparse tanto por elegir buenos valores de hiperparámetros. Por lo general, no necesitamos optimizar Random Forest, ya que el modelo de conjunto es bastante robusto al ruido introducido por los árboles de decisión individuales. El único parámetro que resulta interesante configurar en la práctica es el número de árboles que se eligen para formar el Random Forest

(n_estimators). Normalmente, cuanto mayor sea el número de árboles, mejor será el rendimiento del clasificador de bosques aleatorio a expensas de un mayor coste computacional [67].

De cara a encontrar cual es el numero óptimo de árboles se usa el método GridSearch con RandomForestClassifier(), con valores de n_estimators 10, 50, 100, 200 [72], confirmando que los mejores resultados se obtienen cuando se maximiza n_estimators. Adicionalmente también se configura como hiperparámetro la longitud máxima de cada árbol, que, de forma similar al caso anterior, toma valores desde 6 a 20.

3.4.3 Support Vector Machines

Las maquinas de soporte vectorial o SVM (*Support Vector Machines*) es un método para la clasificación de datos lineales y no lineales. SVM es un algoritmo que funciona de la siguiente manera. Utiliza un mapeo no lineal para transformar los datos de entrenamiento e incrementar su dimensión. Dentro de esta nueva dimensión, busca el hiperplano de separación óptimo lineal (es decir, un "límite de decisión" que separa una clase de otra, en nuestro caso, las tuplas con clase 0 (agua no salina) y con clase 1 (agua salina)). Con un mapeo no lineal apropiado a una dimensión suficientemente alta, los datos de dos clases siempre pueden estar separados por un hiperplano [47], [69].

Como se comentó anteriormente, SVM encuentra hiperplano que maximiza la distancia entre las tuplas más cercanas de ambas clases. Está distancia se llama margen y juega un papel fundamental en este método. Las tuplas que delimitan el hiperplano que maximiza el margen son llamadas máquinas de soporte o *support machines,* que se pueden ver resaltadas en la figura 34 [47], [69].

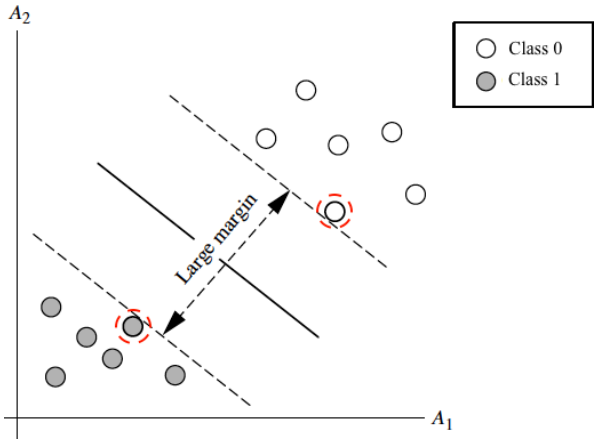


Figura 34: Vectores de soporte y margen en SVM

En este estudio se aplica SVM recurriendo al método SVC (*Support Vector Classification*) de la librería scikit-learnt [73]. De cara a encontrar el mejor clasificador se busca la combinación óptima de hiperparámetros aplicando GridSearch al clasificador SVC. Los hiperparámetros que se exploran en este

caso son gamma y C, que toman valores de 0.001, 0.01, 0.1, 1, 10 y 0.01, 0.1, 1, 10, 50, 100, 200, respectivamente.

Gamma es un hiperparámetro utilizado con SVM no lineales. Uno de los *kernels* no lineales más utilizados es la función de base radial (RBF). El parámetro Gamma de RBF controla la distancia de la influencia de un solo punto de entrenamiento. Los valores bajos de gamma indican un gran radio de similitud que da como resultado que se agrupen más puntos. Para valores altos de gamma, los puntos deben estar muy cerca entre sí para ser considerados en el mismo grupo (o clase). Por lo tanto, los modelos con valores gamma muy grandes tienden al *overfitting* o sobreajuste [74].

C es el parámetro de penalización del error. Esto significa que C agrega una penalización por cada clase mal clasificada. Si C es pequeño, la penalización por puntos mal clasificados es baja, por lo que se elige un límite de decisión con un gran margen a expensas de un mayor número de errores de clasificación. Si C es grande, SVM intenta minimizar el número de muestras mal clasificadas debido a la alta penalización que resulta en un límite de decisión con un margen más pequeño. La penalización no es la misma para todos los ejemplos mal clasificados, sino directamente proporcional a la distancia al límite de decisión. Para valores altos de C existe el riesgo de *overfitting* o sobreajuste [74].

3.4.4 AdaBoost

El Boosting es un método crea un clasificador fuerte a partir de una serie de clasificadores débiles. Esto se hace construyendo un modelo a partir de los datos de entrenamiento, y luego creando un segundo modelo que intenta corregir los errores del primer modelo. Los modelos se añaden hasta que el conjunto de entrenamiento se predice perfectamente o se añade un número máximo de modelos [60].

AdaBoost (*Adaptative Boosting*) es un popular algoritmo de *boosting*. AdaBoost se puede aplicar a cualquier algoritmo de clasificación, por lo que en realidad es una técnica que se basa en otros clasificadores en lugar de ser un clasificador en sí mismo, en este caso se usa el algoritmo SAMME.R, basado en árboles de decisión. A continuación, se explica brevemente la construcción de los clasificadores con AdaBoost.

Dado un conjunto de datos *D* con *d* tuplas etiquetadas por clase, en nuestro caso clase 0 y clase 1, $(X_1, y_1), \ldots, (X_d, y_d)$, donde y_i es la etiqueta de clase de la tupla X_i. Inicialmente, AdaBoost asigna a cada tupla de entrenamiento un peso igual de 1/*d*. Generar *k* clasificadores para crear el conjunto requiere *k* iteraciones a través del resto del algoritmo. En la ronda i, las tuplas de *D* se muestrean para formar un conjunto de entrenamiento, D_i, de tamaño *d*.

La probabilidad de que se seleccione cada tupla se basa en su peso. Un clasificador, M_i, se deriva de las tuplas de entrenamiento de D_i. Su error se calcula usando D_i como un conjunto de test. Los pesos de las tuplas del conjunto de entrenamiento se ajustan de acuerdo con cómo se clasifican. Si

una tupla se clasifica incorrectamente, su peso se incrementa. Si una tupla se clasifica correctamente, su peso se reduce. El peso de una tupla refleja como de difícil es clasificar dicha tupla: cuanto mayor sea el peso, más veces se ha clasificado erróneamente. Estos pesos se utilizarán para generar las muestras de entrenamiento para el clasificador de la siguiente ronda. La idea básica es que cuando construimos un clasificador, es necesario que se centre más en las tuplas mal clasificadas de la ronda anterior. Los distintos clasificadores realizarán mejor la tarea de clasificación en unas tuplas que en otras. De esta manera, se construirán una serie de clasificadores que se complementen entre ellos.

Para realizar la predicción, se asigna un peso al voto de cada clasificador, en función del rendimiento del mismo. Cuanto menor sea la tasa de error de un clasificador, más preciso será y, por lo tanto, mayor será su peso para la votación.

En este estudio se emplea AdaBoost recurriendo al método AdaBoostClassifier() de la librería scikit-learnt [75], que crea un clasificador que será entrenado con el training set definido en apartados anteriores.

Como se ha comentado, AdaBoost crea un clasificador fuerte a través de un conjunto de clasificadores débiles. El número de clasificadores débiles puede ser definido a través del parámetro n_estimators. Adicionalmente, en este estudio también se recurre al parámetro learning_rate, que controla la contribución de cada estimador débil a la predicción conjunta. Valores altos o bajo serán apropiados dependiendo del número de modelos usados en el conjunto. Se debe fijar un equilibrio entre la contribución de los modelos y el número de árboles en el conjunto. Más árboles pueden requerir un learning_rate menor y menos árboles pueden requerir un learning_rate mayor [66].

Con el fin de buscar la configuración óptima de hiperparámetros, se aplica GridSearch sobre AdaBoostClassifier(). Los hiperparámetros n_estimators y learning_rate se evalúan con los valores 10, 50, 100, 200 y 0.01, 0.1, 1, 2, respectivamente. Al igual que en Random Forest, donde también se trabaja con estimadores débiles, se obtiene una mayor precisión de clasificación cuando se usa un número elevado de estimadores débiles.

3.4.5 Redes Neuronales Artificiales (ANN)

En general, una red neuronal artificial es un conjunto de unidades de entrada / salida conectadas en las que cada conexión tiene un peso asociado. Durante la fase de aprendizaje, la red aprende ajustando los pesos para poder predecir la etiqueta de clase correcta del conjunto de variables de entrada [68].

Existen distintos algoritmos para realizar el proceso de aprendizaje en una red neuronal, aunque el más utilizado es la propagación hacia atrás o *backpropagation* (por su término en inglés). Este algoritmo aprende de forma iterativa un conjunto de pesos para predecir la etiqueta de clase del conjunto de variables de entrada en una red neuronal multicapa alimentada hacia adelante

(*feed-forward* en inglés), que es el tipo de red neuronal usada en este estudio (ver figura 34) [47].

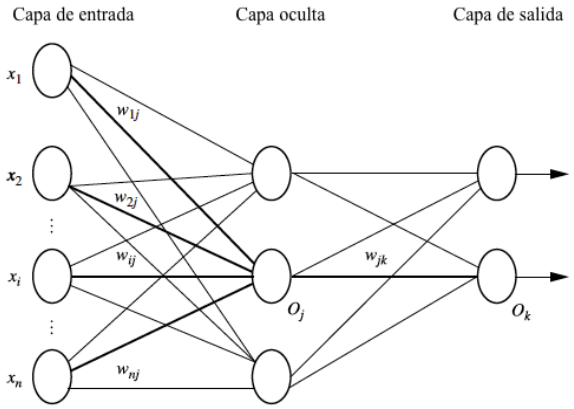


Figura 34: Red multicapa *feed-forward*

Cada capa está formada por unidades o neuronas. Las entradas a la red corresponden a los atributos o variables del conjunto de entrenamiento. Estas entradas pasan a través de la capa de entrada, luego se les atribuye un peso y se introducen a una segunda capa de unidades, conocida como capa oculta. Las salidas de las unidades de capa oculta se podrían introducir en otra capa oculta, y así sucesivamente. El número de capas ocultas es arbitrario, aunque en la práctica, normalmente sólo se utiliza una, como en este estudio. Las salidas ponderadas de la última capa oculta se introducen en las unidades que componen la capa de salida, que emite la predicción de la red para las variables determinadas [68].

Cada unidad de salida (neuronas de la capa oculta y capa de salida) toma, como entrada, una suma ponderada de las salidas de las unidades de la capa anterior. Esta aplica una función no lineal (activación) a la entrada ponderada. Este tipo de red neuronal es capaz de modelar la predicción de clase como una combinación no lineal de las entradas. Desde un punto de vista estadístico, realizan regresiones no lineales. Las redes neuronales vistas en este apartado, con suficientes unidades ocultas y suficientes muestras de entrenamiento, pueden aproximarse mucho a cualquier función [47].

Como se ha comentado, en este estudio se recurre a una arquitectura multicapa con 20 unidades de entrada, una capa oculta y 2 unidades de salida. Para formar esta arquitectura se recurre a los métodos Sequential y Dense. Una vez esté formada la red neuronal se crea un clasificador con el método KerasClassifier() que será entrenado con el conjunto de entrenamiento descrito anteriormente. Todos estos métodos están contenidos en la librería Keras, que se ejecuta sobre TensorFlow [76].

En este caso utilizando GridSearch con KerasClassifier*()* se explorarán los siguientes hiperparámetros:

- Units: Número de neuronas en la capa oculta. Tomará los valores de 10,30 y 40.

- Learning_rate: es un hiperparámetro que controla cuánto cambia el modelo en respuesta al error estimado cada vez que se actualizan los pesos del modelo. Un valor demasiado pequeño puede resultar en un proceso de entrenamiento largo, mientras que un valor demasiado grande puede resultan en un aprendizaje de pesos no óptimo o inestable. En este caso, los valores de learning_rate que se estudian son, 0.1, 0.5 y 0.9 [60].
- Epochs: es un hiperparámetro que define el número de veces que el algoritmo de aprendizaje pasa a través de todo el conjunto de entrenamiento. El número de epochs suele fijarse a valores altos, para los que el modelo presenta mejores resultados de clasificación [60]. En este estudio los valores usados son, 50,75 y 100.

3.5 Análisis de los resultados

En este capítulo se analizarán los resultados obtenidos tras la aplicación de los distintos métodos de aprendizaje automático descritos anteriormente, de cara a tratar de dar respuesta a las preguntas que se lanzaron al principio de este estudio.

La tabla 2 muestra la precisión de clasificación de la variable *Salinidad_mezcla* hallada para cada método de aprendizaje automático, en los conjuntos de entrenamiento y validación, y para cada método de imputación de *missing values*, analizados en el apartado 3.3.2.1. Los mejores resultados de precisión se observan cuando se recurre a Random Forest como método de aprendizaje automático y se usa interpolación lineal como estrategia de gestión de *missing values*, alcanzando un 93,94% de precisión de clasificación.

Adicionalmente, a nivel de estrategia de gestión de *missing values,* también se puede afirmar que con interpolación lineal se obtienen los mejores resultados de precisión de clasificación independientemente del método de aprendizaje automático que se use.

También se obtienen buenos resultados para todos los métodos de aprendizaje automático usando la interpolación Akima. Analizando el apartado 3.3.2.1 se puede ver que usando interpolación lineal como método de imputación logramos conservar mejor la distribución estadística de la variable *Salinidad_mezcla* tras la imputación de los *missing values*, por lo tanto, se puede concluir que el método más adecuado para gestionar los *missing values* en este estudio es la interpolación lineal.

Imputation Method	Decision Trees	Random Forest	Support Vector Machine	AdaBoost	ANN
Drop NAN	87.26	89.46	87.26	87.96	85.26
Mean	88.32	88.97	88.16	86.82	86.21
Median	88.36	90.36	88.45	88.53	87.83
Rolling Mean	89.74	92.56	88.61	90.04	83.70
Rolling Median	89.52	92.82	89.61	89.43	83.00
Linear Interpolation	91.21	93.94	91.01	91.37	86.66
Quadratic Interpolation	89.14	91.89	88.73	88.61	82.75
Cubic Interpolation	89.06	92.05	88.12	89.06	82.30
Akima Interpolation	91.27	93.85	90.62	91.11	85.87
Spline Interpolation	89.83	92.15	87.18	88.04	81.61
Linear Regressor Predictor	87.63	89.79	88.32	88.45	86.41

Tabla 2: Precisión de clasificación en los conjuntos de entrenamiento y validación

Siguiendo con el análisis de la tabla 2, se puede observar que los peores resultados de precisión se alcanzan cuando se eliminan los *missing values* del conjunto de datos. Este resultado es lógico ya que se está eliminando una gran cantidad de información, 1720 registros exactamente, con la que entrenar los modelos de aprendizaje automático, por lo que no es recomendable recurrir a este método de gestión de *missing values* en un caso como el de este estudio.

También se puede observar en la tabla 2 que el método de aprendizaje automático que peor precisión de clasificación presenta es la red neuronal (ANN). Como se muestra en la figura 35, las redes neuronales presentan mejores resultados que los métodos tradicionales de aprendizaje automático (como arboles de decisión, Random Forest, etc) cuando se dispone de un conjunto de datos extenso [60]. Por lo tanto, si se quisiera mejorar la precisión de clasificación usando redes neuronales, se debería incrementar el conjunto de datos.

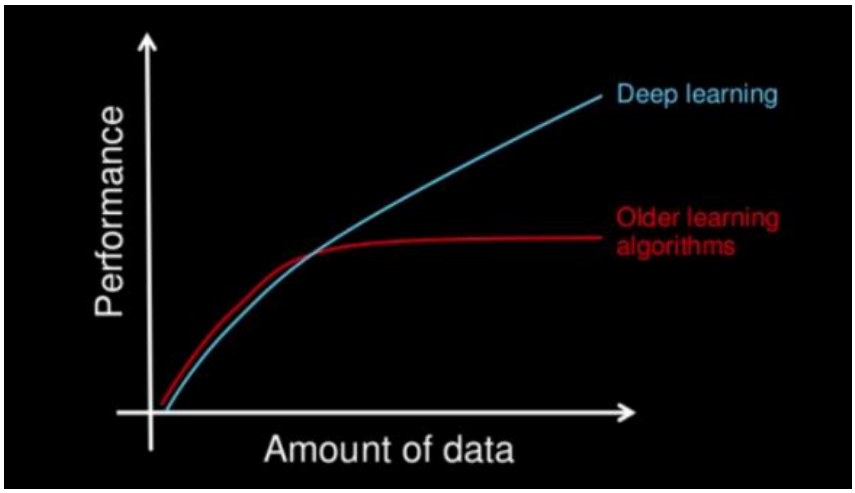


Figura 35: Deep learning vs métodos tradicionales [60]

También se obtiene la precisión de clasificación para el conjunto de test, como se puede ver en la tabla 3. Como se comentó en apartados anteriores, el conjunto de test no se usa en el proceso de entrenamiento del modelo de clasificación, por lo que analizando estos resultados se sabrá si el modelo obtenido es capaz de generalizar correctamente para conjuntos de datos nuevos. En esta ocasión, nuevamente, el mejor resultado, 95.17%, se observa para el método Random Forest y usando interpolación lineal para la gestión de los *missing values*. En el mismo sentido y al igual que en la tabla 2, interpolación lineal es el método de gestión de *missing values* con el que se obtienen mejores resultados en todos los métodos de aprendizaje automático.

Imputation Method	Decision Trees	Random Forest	Support Vector Machine	AdaBoost	ANN
Drop NAN	84.18	88.70	85.31	89.26	81.35
Mean	86.89	87.81	86.66	86.43	85.51
Median	84.82	90.11	88.96	87.81	87.81
Rolling Mean	89.68	92.13	87.96	88.20	82.55
Rolling Median	89.92	94.10	89.92	87.96	80.83
Linear Interpolation	91.26	95.17	93.10	93.10	87.58
Quadratic Interpolation	89.32	92.80	87.47	90.48	85.38
Cubic Interpolation	88.63	93.50	88.16	89.55	84.91
Akima Interpolation	90.95	94.89	89.32	92.11	87.00
Spline Interpolation	89.88	93.33	87.81	88.96	84.13
Linear Regressor Predictor	88.96	90.57	87.12	88.50	86.20

Tabla 3: Precisión de clasificación en los conjuntos de test

Siguiendo el análisis sobre el conjunto de test, en la figura 36 se muestra la matriz de confusión tras aplicar el clasificador obtenido, a través de Random Forest y aplicando interpolación lineal para imputación de *missing values*, al conjunto de test. Analizando esta matriz de confusión se obtienen las siguientes observaciones:

- Se clasifican correctamente como muestras no salinas (*Salinidad_mezcla* = 0) 201 muestras.
- Se obtienen 9 falsos negativos. Esto significa que se predicen 9 muestras como no salinas (*Salinidad_mezcla* = 0) cuando en realidad se deberían haber clasificado como salinas (*Salinidad_mezcla* = 1).
- Se obtienen 11 falsos positivos. Esto significa que se predicen 11 muestras como salinas (*Salinidad_mezcla* = 1) cuando en realidad se deberían haber clasificado como no salinas (*Salinidad_mezcla* = 0).
- Se clasifican correctamente como muestras salinas (*Salinidad_mezcla*=1) 214 muestras.

```
#La matriz de confusion quedaría
confusion_matrix(y_test_gs, y_pred_rf)
```

```
array([[201,   9],
       [ 11, 214]])
```

Figura 36: Matriz de confusión del clasificador Random Forest

Una vez ha quedado claro que el método de gestión de *missing values* mas adecuado es la interpolación lineal, en las siguientes figuras es posible ver como funciona este método a la hora de imputar valores a los *missing values.* En la figura 37 y figura 38 se puede ver los valores de la variable *Salinidad_mezcla* antes y después del uso de interpolación lineal para la imputación de *missing values,* seleccionando el año 2012 como muestra*.*

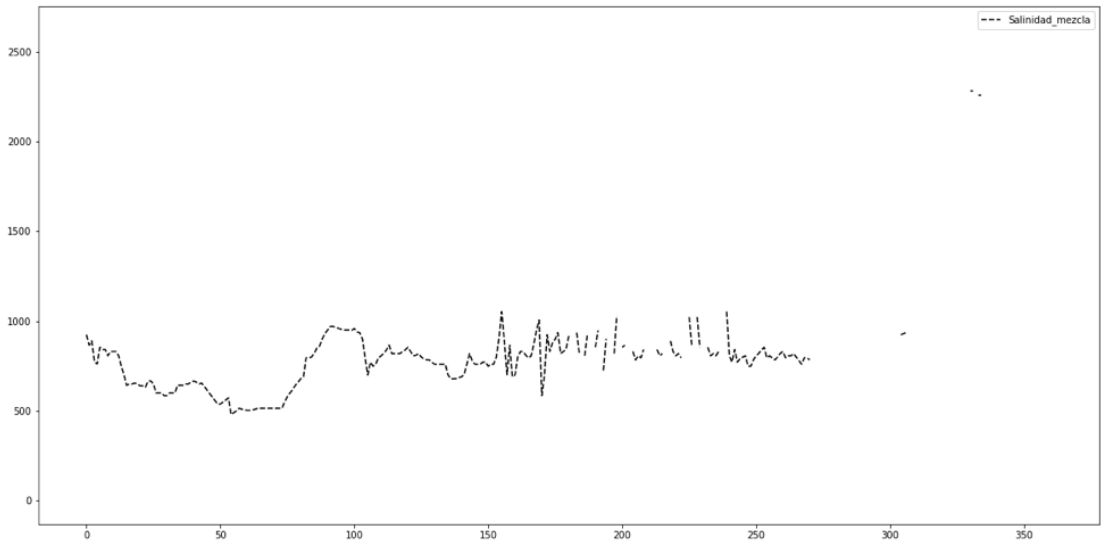


Figura 37: Valores de *Salinidad_mezcla* para el año 2012

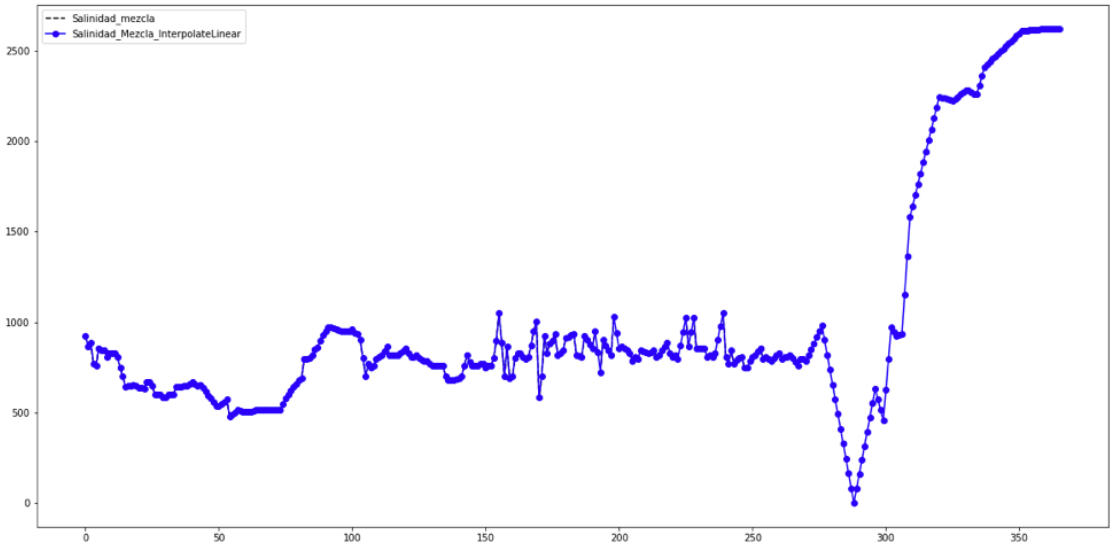


Figura 38: Valores de *Salinidad_mezcla* con interpolación lineal para el año 2012

Se ha visto en las tablas anteriores que el método de aprendizaje automático que mejor resultados presenta es Random Forest, con precisiones de hasta 93.94% para el conjunto de entrenamiento y validación, y 95.17% para el conjunto de test. Para alcanzar estos valores se ha sometido al conjunto de datos a una limpieza y preprocesado, según se ha descrito en apartados anteriores y, además, se ha llevado a cabo un proceso de *tunning* de los hiperparámetros asociados al algoritmo Random Forest mediante el uso del método GridSearch. Para aplicar GridSearch se seleccionan distintos valores de los hiperparámetros asociados al método Random Forest, n_estimators y max_depth. El hiperparámetro n_estimators toma valores 10, 50, 100 y 200 y max_depth toma valores dentro del rango [6,20). A partir de GridSearch se entrena el modelo creado a través del método RandomFortestClassifier() con una combinación diferente de hiperparámetros, en la figura 39 se puede ver la precisión de clasificación que se obtiene con el conjunto de entrenamiento para todas las combinaciones de hyperparámetros. Como ya se comentó en el apartado anterior, a mayor número de estimadores y más profundidad de los arboles, más precisión se obtendrá.

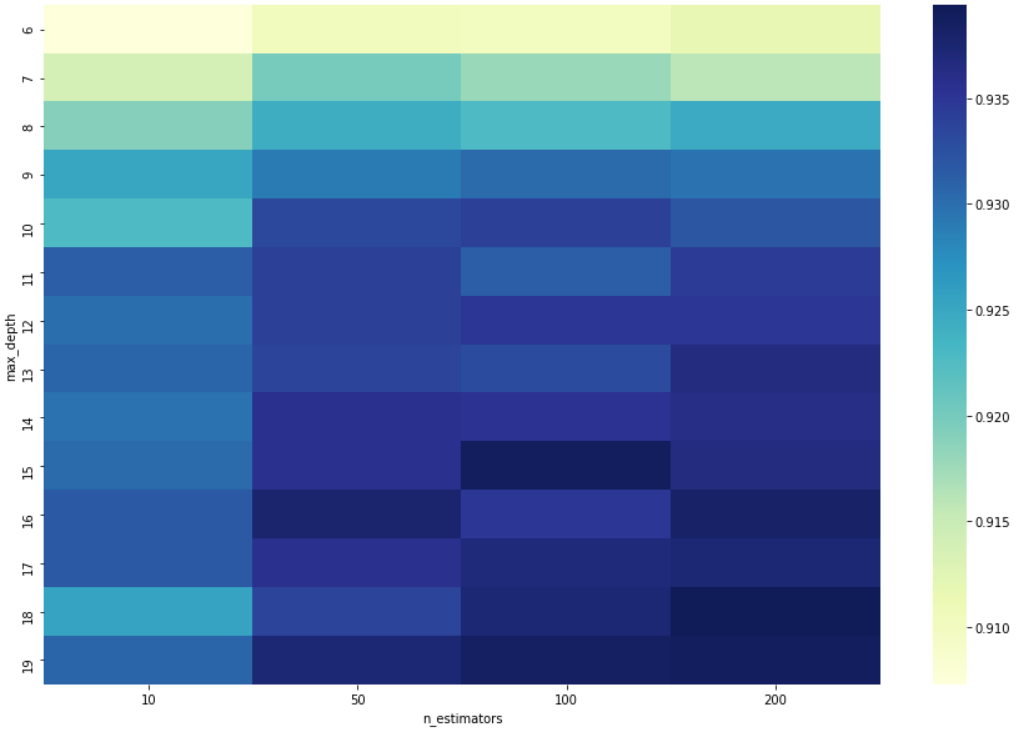


Figura 39: Tunnig de Hiperparámetros para RandomForestClassifier()

Una vez se ha obtenido el método de aprendizaje automático óptimo y el método de gestión de *missing values* adecuado, es posible realizar algún análisis sobre los parámetros de interés del sistema de presas.

Según se vio en los objetivos inicialmente expuestos, la Junta de Andalucía estaba interesada en relacionar la variable *Salinidad_mezcla* con el resto de variable de explotación. A continuación, se relaciona la variable *Salinidad_mezcla* con las variables *Salinidad_guadalhorce* y *Cota_guadalhorce*, ambas del interés de la Junta de Andalucía. Así, en la figura 40 es posible ver las fronteras de decisión que Radom Forest utiliza para realizar la clasificación entre mezcla de aguas salina (*Salinidad_mezcla*=1) y mezcla de aguas no salina (*Salinidad_mezcla*=0). Es decir, esta figura muestra, en función de los valores de *Salinidad_guadalhorce* y *Cota_guadalhorce*, como puntos rojos aquellas muestras clasificadas como mezcla salina (*Salinidad_mezcla*=1) y como puntos azules aquellas muestras clasificadas como mezcla no salina (*Salinidad_mezcla*=0).

Es posible extraer algunas conclusiones de esta figura, ya que existen franjas de clasificación claramente marcadas, como la franja correspondiente a valores bajos de *Cota_guadalhorce* (*Cota_guadalhorce* < 0.3). En esta franja, la variable *Salinidad_mezcla*, se clasifica siempre como mezcla de aguas salina independientemente del valor de *Salinidad_guadalhorce*. También es posible ver en la pendiente de la gráfica que la *Salinidad_guadalhorce* baja progresivamente cuando *Cota_guadalhorce* sube. Así, para valores altos de *Cota_guadalhorce* (*Cota_guadalhorce* ≈ 1) *Salinidad_mezcla* siempre se clasifica como no salina. Se puede concluir que cuando en la presa Guadalhorce se observan valores medios y altos de su cota y valores medios y altos de su salinidad, es más probable que la mezcla sea no salina. A

continuación, se analizará la influencia de cada variable en la salinidad de la mezcla de forma individual.

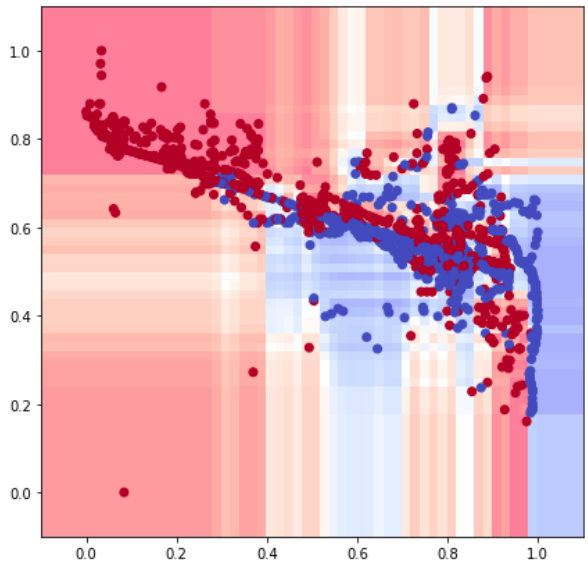


Figura 40: Fronteras de decisión de RandomForestClassifier

Para continuar analizando la influencia de variables de interés en la salinidad de la mezcla se pretende hacer variar una variable y mantener el resto de variables con valores constantes, al valor de su mediana, de cara a ver como varían las predicciones de *Salinidad_mezcla*. Para ello se decide crear un dataset con 102 registros y las 20 variables con las que se entrenó el modelo original. Notar varios puntos antes de comenzar el análisis:

- Los resultados en ejecuciones futuras pueden variar dada la naturaleza estocástica del algoritmo o las diferencias en la precisión numérica.
- Variar una variable, entre su mínimo y su máximo, y mantener las variables restantes con valores constantes, debe verse únicamente como un ejercicio de análisis, pero nunca como un resultado vinculante, ya que se está alterando la naturaleza de los datos.
- Se usa la medina para asignar un valor constante a las distintas variables ya que es menos sensible a outliers y valores extremos que la media.

En el primer análisis, los 102 elementos de la variable *Salinidad_guadalhorce* variará en saltos de 40 entre el valor mínimo (3.194) y máximo (3990). Las 19 variables restantes contendrán los 102 elementos idénticos, con el valor de su mediana.

En la figura 41 se puede ver el resultado de este análisis. Se observa que a partir del valor 2500 p.p.m se comienza a clasificar como salina la mezcla de aguas. Este resultado es interesante ya que permite acotar el valor de la salinidad de la mezcla en función de los valores de salinidad de la presa Guadalhorce.

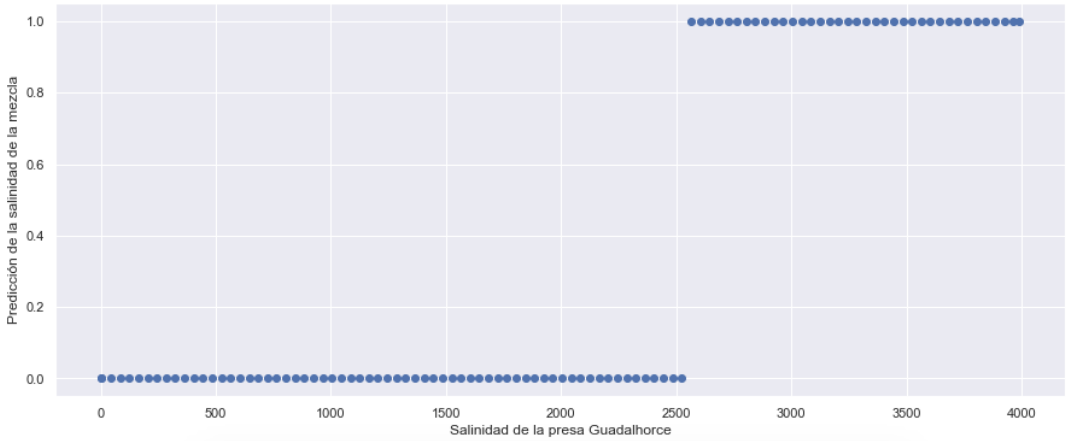


Figura 41: Predicción de *Salinidad_mezcla* en función de *Salinidad_guadalhorce*

Para evaluar las predicciones de *Salinidad_mezcla* en función de los valores de *Cota_guadalhorce* se realizará el mismo ejercicio que en el caso anterior. Por lo tanto, creamos un dataset con 20 variables, donde *Cota_guadalhorce* tomará 102 valores entre su valor mínimo (348.6 m) y su valor máximo (362.8 m) con saltos de 0.135 m. Las 19 variables restante permanecerán con un valor constante, al valor de su mediana. Analizando la figura 42 se puede ver, como ya pudo intuirse en la figura 40, que las muestras se predicen como salinas para valores bajos de la cota (desde 348.6 m hasta 354.2 m, aproximandamente) y como no salinas para valores altos (desde 354.2 m hasta 362.8 m, aproximadamente). Esto puede tener sentido ya que la cota suele subir cuando existen lluvias, por lo tanto, la salinidad de la presa de Guadalhorce y, en consecuencia, la salinidad de la mezcla, puede bajar.

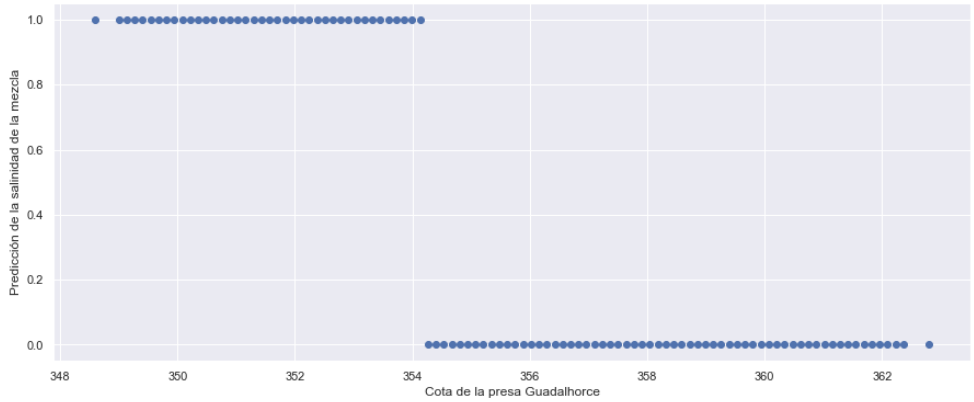


Figura 42: Predicción de *Salinidad_mezcla* en función de *Cota_guadalhorce*

También es posible analizar la predicción de *Salinidad_mezcla* en función de las variables relacionadas con el volumen de agua de las distintas presas. Por ejemplo, en la figura 43 se puede observar la predicción de la variable *Salinidad_mezcla* en función del volumen total de las tres presas principales, Guadalhorce, Guadalteba y Conde del Guadalhorce. El clasificador óptimo hallado clasifica las muestras como salinas si el volumen total de las tres presas es inferior a 208 hm^3.

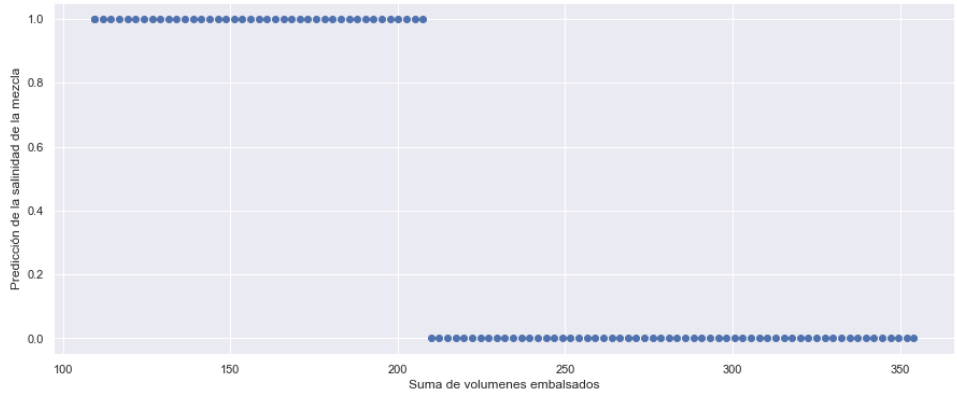


Figura 43: Predicción de *Salinidad_mezcla* en función de la suma de los volúmenes embalsados de Guadalhorce, Guadalteba y Conde del Guadalhorce

En apartados anteriores se definió que la mezcla de aguas, usada para regadío y consumo, se obtenía principalmente a través del desembalse de las presas Guadalhorce y Guadalteba, y mezclando dichas aguas desembalsadas. El agua resultante muestra una determinada salinidad, que en este estudio se ha llamado *Salinidad_mezcla*. Así, es interesante saber como influyen los desembalses de sendas presas en la salinidad de la mezcla de aguas.

De la figura 44 se deriva que, excepto en un pequeño intervalo de [0,0.05) hm^3, cuando se usa cualquier cantidad de agua desembalsada de la presa Guadalteba, manteniendo el resto de variables con un valor constante, las muestras se clasifican como no salinas. Esto tiene sentido ya que el agua de la presa Guadalteba es de muy buena calidad y de una salinidad muy baja.

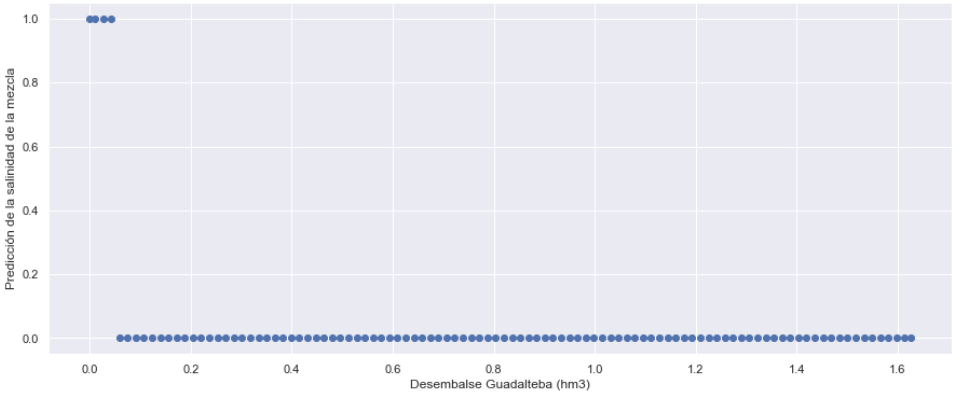


Figura 44: Predicción de *Salinidad_mezcla* en función del desembalse de Guadalteba

En el mismo sentido que el análisis anterior, en la figura 45 se muestra la predicción de la variable *Salinidad_mezcla* en función de distintos valores de desembalse de la presa Guadalhorce, manteniendo el resto de variable con valores constantes. Notar que la variable *Volumen_desembalse_guadalhorce* no es seleccionada como una variable relevante, según el método de reducción de dimensionalidad descrito en el apartado 3.3.4, por lo tanto, únicamente para realizar este análisis, se seleccionaron 40 variables en lugar de 20, de cara a que la variable *Volumen_desembalse_guadalhorce* apareciera entre las variables seleccionadas.

El resultado es que, excepto si se usa un volumen menor a 0.0935 hm^3 para producir la mezcla de aguas, las muestras serán clasificadas como salinas. Este resultado tiene sentido, ya que el agua de la presa Guadalhorce es muy salina y si se mantienen el resto de variables constantes, hay una gran probabilidad de que el agua salina del Guadalhorce aporte una gran salinidad a la mezcla de aguas.

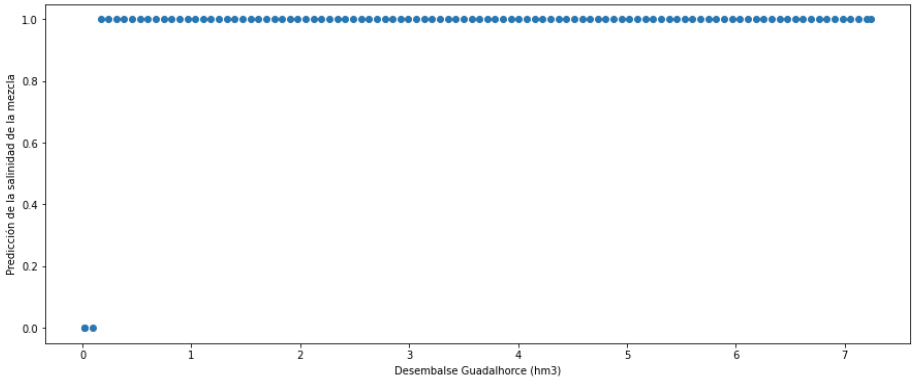


Figura 45: Predicción de *Salinidad_mezcla* en función del desembalse de Guadalhorce

4. Conclusiones y líneas de futuro

Para concluir, en este capitulo se presenta un resumen con las principales conclusiones, respondiendo a las preguntas planteadas inicialmente e incluyendo líneas de trabajo futuras.

4.1 Conclusiones

Una de las partes mas importantes en un proyecto de ciencia de datos es el trabajo con el *stakeholder* o interesado, en este caso, la Junta de Andalucía. En este estudio siempre se ha seguido una comunicación fluida y transparente con el objetivo de que se supiera en todo momento en que punto se encontraba el proyecto. La colaboración conjunta ha permitido que se fueran alcanzando los objetivos del proyecto según estaban proyectados, ya que su consecución dependía en gran medida del hecho de poder disponer de los datos y poder entender la labor de explotación en el sistema de presas del valle del Guadalhorce.

El entender cúal es el proceso de negocio asociado a un *stakeholder* es primordial para poder plantear los objetivos adecuados y gestionar las expectativas. En este caso y como se indicó anteriormente, entender el funcionamiento del sistema de presas y la labor de explotación del personal de la Junta de Andalucía se tradujo en la puesta en claro de una serie de objetivos y el posterior análisis cualitativo de los resultados.

Antes relacionar los resultados obtenidos con los objetivos planteados inicialmente se debe entender que en este estudio no se presenta un problema de optimización multivariable, que permitiría maximizar o minimizar unas variables en función de otras. En problemas en los que se busca una clasificación mediante aprendizaje supervisado, no suele abordarse una optimización multivariable. Lo que se ha realizado en este estudio, como ya se ha desarrollado en este documento, es el desarrollo de un modelo de aprendizaje automático que optimiza la predicción muestras de agua en salina o no salina en función de un conjunto amplio de variables.

Considerando lo planteado anteriormente y relativo al objetivo principal relacionado con encontrar una mezcla óptima de agua, se han realizados distintos análisis que han permitido conocer un poco más con que parámetros se puede trabajar para optimizar dicha mezcla de aguas. En primer lugar, se han encontrado 20 parámetros que influyen de forma significativa en la salinidad de la mezcla de aguas (ver figura 30).

Dentro de estos parámetros se ha observado que la salinidad del Guadalhorce influye en gran medida en la salinidad de la mezcla. Así, según el modelo hallado, si se utiliza agua del Guadalhorce con una salinidad mayor a 2500 p.p.m, la mezcla de aguas, utilizada para regadíos y consumo, será salina, esto es que tendrá una salinidad mayor a 900 p.p.m. A la inversa, si utilizamos agua del Guadalhorce con una salinidad menor a 2500 p.p.m, con una probabilidad

alta, la mezcla de aguas no será salina, esto es que la mezcla tendrá una salinidad menor o igual a 900 p.p.m.

Adicionalmente, también se ha visto que, según el modelo obtenido, si se utiliza mas de 0.05 hm^3 de agua desembalsada procedente de la presa Guadalteba y menos de 0.0935 hm^3 de agua desembalsada de la presa Guadalhorce, se obtendrá, con una probabilidad alta, una mezcla de agua no salina. Por lo tanto, se puede afirmar que, según el modelo obtenido, el máximo volumen de agua que se puede utilizar de la presa Guadalhorce manteniendo la mezcla no salina es 0.0935 hm^3. De la misma forma, según modelo, el mínimo volumen de agua desembalsada de la presa Guadalteba que mantiene la mezcla no salina es 0.05 hm^3.

Uno de los objetivos secundarios era hallar la influencia la cota de la presa de Guadalhorce en la salinidad de la mezcla de aguas. Por un lado, de forma cualitativa, se ha visto que a medida que aumenta la cota de la presa Guadalhorce, baja la salinidad de dicha presa. Es decir, que cuando la cota es alta, la mezcla de aguas tiene mas probabilidad de ser no salina que salina. Por otro lado, de forma cuantitativa, como se ve en la figura 42, según el modelo obtenido, si la cota de la presa Guadalhorce presenta una altura mayor de 354.2 m, la mezcla tendrá una alta probabilidad de ser no salina. Por el contrario, si dicha cota está por debajo de 354.2 m, con la misma probabilidad, la mezcla de aguas será salina.

Por lo tanto, se puede concluir que los objetivos planteados inicialmente se han logrado satisfacer, sabiendo que se podrían mejorar si se realizara un proceso de optimización multivariable.

Como ya se ha comentado en el desarrollo del estudio, un factor que ha marcado y, en mayor o menor medida, sesgado los resultados obtenidos es el gran número de *missing values* presentes en los datos en bruto proporcionados por la Junta de Andalucía. Así, se ha visto que muchos de los métodos de aprendizaje automático más usados, o bien, no se pueden aplicar directamente a conjuntos de datos con *missing values,* o bien, no presentan buenos resultados, por lo tanto, y para mitigar la pérdida de información, en lugar de eliminar estos *missing values*, es mejor completar los registros faltantes con valores apropiados y para ello se han estudiado distintos métodos de imputación.

También ha quedado patente la ineficiencia que se introduce en la etapa de limpieza de datos cuando la información se encuentra dispersa y almacenada en un formato inadecuado. Las consecuencias de acometer un proyecto con los datos en estas condiciones son múltiples; dedicación de una ingente cantidad de tiempo, pérdida de información por la manipulación de los datos, pérdida de contexto de los datos, brechas de seguridad, etc.

Por último, en relación con los dos puntos anteriores, un factor que afecta directamente a la existencia de *missing values* y la dispersión de los datos es el hecho de que los datos, en su mayoría, sean tomados y almacenados de forma

manual, lo que introduce error y complejidad considerable, patente de forma práctica.

4.2 Líneas futuras

En base a todo lo expuesto durante este estudio, a continuación, se presentan una serie de líneas de trabajo futuras que podría ayudar a mejorar los resultados obtenidos.

Uno de los enfoques que podrían ayudar a mejorar la precisión de clasificación de los modelos de aprendizaje automático obtenidos es el apilamiento o *stacking* (por su término en inglés) de modelos de aprendizaje automático. La idea principal del *stacking* es utilizar predicciones de modelos de aprendizaje automático del nivel anterior como variables de entrada para los modelos del siguiente nivel. Este tipo de enfoque está siendo usado de forma extensa en los últimos tiempos, por ejemplo, es muy popular entre los participantes de la comunidad de Kaggle [77]. En varios estudios se puede ver que, al utilizar modelos de *stacking* multinivel, se pueden obtener resultados más precisos en comparación con los modelos individuales [78].

Otra tendencia en los proyectos de aprendizaje automático es el uso de PyCaret, que es una librería Python que incluye modelos pre-entrenados optimizados para determinadas tareas, de forma que permite evaluar, comparar y ajustar modelos en un conjunto de datos determinado con solo unas pocas líneas de código [79]. La idea sería usar PyCaret con el conjunto de datos presentados en este documento con el fin de comparar los resultados obtenidos con un método alternativo que introduzca mayor grado de automatización en la fase de contracción de modelos.

Relativo a uno de los objetivos del estudio, maximizar variables minimizando otras es un problema cuya solución óptima se podría encontrar aplicando métodos de optimización multivariable, que implicaría encontrar la ecuación o ecuaciones no lineales que maximicen unas variables y minimicen otras. Por lo tanto, el estudio descrito en este documento se podría completar con un estudio de optimización multivariable.

Las líneas futuras comentadas anteriormente van muy enfocadas a mejorar los resultados obtenidos en el estudio presentado en este documento, sin embargo, se hace necesario mencionar algunas aproximaciones que pudieran mejorar la calidad de los datos y automatizar algunos de los procesos con el objetivo de reducir fallos en la toma de datos.

Uno de los planteamientos con los que está comenzando a trabajar la Junta de Andalucía es la implantación de servidores Scala. Estos servidores están conectados con distintos elementos de explotación del sistema de presas, casetilla de meteorología, sensores de auscultación, SAIH, etc. Estos servidores permiten incorporar enfoques Big Data para procesar en tiempo real los datos que se estén recogiendo, así como aplicar técnicas de ETL y procesos de visualización. Este tipo de servidores aún está en proceso de prueba en el sistema de presas del valle del Guadalhorce, por lo que, se

propone realizar un análisis de datos una vez los servidores Scala se encuentren en producción y hayan podido recoger suficientes datos. Esto ayudará a resolver el problema de falta de datos, formato y coherencia, así como la inclusión de análisis predictivos y visualizaciones en tiempo real.

Relacionado con el formato de los datos, como se ha visto durante el desarrollo de este estudio, actualmente los datos se almacenan en ficheros Excel, por lo que la información queda dispersa, difícilmente accesible y vulnerable. Como alternativa, se podría considerar implantar un almacén de datos o *Data Warehouse*, que es un tipo de bases de datos que está orientado a almacenar datos con el objetivo de optimizar las consultas y generar informes, agregando y resumiendo datos de diferentes fuentes. Suelen alimentarse de las diferentes bases de datos usadas en una organización y permiten un acceso centralizado con el objetivo de poder realizar consultas más complejas y eficientes [43].

Por último, y con el objetivo de garantizar la calidad y coherencia de los datos se propone implantar mecanismos de gestión de datos o *Data Governance* según describe el modelo DAMA (*Data Managment Agency*), de cara a cubrir las áreas indicadas en la figura 46 [80].

Figura 46: Areas de *Data Governance* según DAMA

5. Glosario

NaCl	Cloruro Sódico
Na	Sodio
SO_4	Sulfato
Mg	Magnesio
Ca	Calcio
K	Potasio
ML	Machine Learning
ANN	Artificial Neural Network
MLR	Multiple Linear Regression
RF	Random Forest
kNN	k-Nearest Neighbour
SGD	Stochastic Gradient Descent
IWG	Irrigation Water Quality
SAR	Sodium Absorption Ratio
RAS	Ratio de Absorción de Sodio
TDS	Total Dissolved Salts/Solids
CNN	Convolutional Neural Network
WQI	Water Quality Index
SAIH	Sistema Automático de Información Hidrológica
hm^3	Hectómetros cúbicos
m	Metro ó metros
p.p.m	Partes por millón

mm Milímetro. En el caso de la unidad de medida para las precipitaciones, litro de lluvia caída en un metro cuadrado

ANOVA Analysis of Variance

RBF Radial Basis Function

6. Bibliografía

[1] https://www.waterquality.gov.au/issues/salinity

[2]https://www.water.wa.gov.au/water-topics/water-quality/managing-water-quality/understanding-salinity

[3] **Munns R** (2002) *Comparative physiology of salt and water stress.* Plant Cell Environ. 25, 239-250.

[4] **Ghassemi F., Jakeman A.J., Nix H.A.** (1995). *Salinisation of land and water resources: Human causes, extent, management and case studies.* UNSW Press, Sydney, Australia, and CAB International, Wallingford, UK.

[5] **Ghassemi et al.** (1995) *compiled from FAO data for 1987*

[6] **Munns, R., & Termaat, A.** (1986). *Whole-Plant Responses to Salinity.* Functional Plant Biology, 13(1), 143-160.

[7] **Munns, R., & Tester, M.** (2008). *Mechanisms of salinity tolerance.* Annual Review of Plant Biology, 59, 651-681.

[8] **Scheelbeek, P.F.; Chowdhury, M.A.; Haines, A.; Alam, D.S.; Hoque, M.A.; Butler, A.P.; Khan, A.E.; Mojumder, S.K.; Blangiardo, M.A.; Elliott, P.; et al.** (2017). *Drinking water salinity and raised blood pressure: evidence from a cohort study in coastal Bangladesh*. Environ. Health Pers., 125, 057007.

[9] **Müller, D.N.;Wilck, N.; Haase, S.; Kleinewietfeld, M.; Linker, R.A.** (2019). *Sodium in the microenvironment regulates immune responses and tissue homeostasis.* Nat. Rev. Immunol. 2019.

[10] **Islam, A.; Majumder, A.** (2012). *Hypertension in Bangladesh: A review.* Indian Heart J. 6403, 319–323.

[11] **Jabed, M.; Paul, A.; Nath, T.** (2018). *Peoples' perception of the water salinity impacts on human health: A case study in south-eastern coastal region of Bangladesh.* Expo. Health 2018.

[12]https://www.mayoclinic.org/es-es/diseases-conditions/preeclampsia/symptoms-causes/syc-20355745#:~:text=La%20preeclampsia%20es%20una%20complicaci%C3%B3n,presi%C3%B3n%20arterial%20hab%C3%ADa%20sido%20normal.

[13] **Khan, A.E.; Scheelbeek, P.F.D.; Shilpi, A.B.; Chan, Q.; Mojumder, S.K.; Rahman, A.; Haines, A.; Vineis, P.** (2014). *Salinity in drinking water and the risk of (pre)eclampsia and gestational hypertension in coastal Bangladesh: A case-control study*. PLoS ONE, 9, e108715.

[14] **Nishida, C.; Uauy, R.; Kumanyika, S.; Shetty, P.** (2002). *The Joint WHO/FAO Expert Consultation on diet, nutrition and the prevention of chronic diseases: Process, product and policy implications.* Public Health Nutr. 7, 245–250.

[15] **NHMRC, NRMMC** (2011) *Australian Drinking Water Guidelines Paper 6 National Water Quality Management Strategy*. National Health and Medical Research Council, National Resource Management Ministerial Council, Commonwealth of Australia, Canberra.

[16] **Cnx Bio Español.**(2016) *Conceptos de Biología*. OpenStax CNX. http://cnx.org/contents/e7a016d3-91fc-4ba0-9e05-a33e986f3d94@12.1.

[17] **Kefford BJ.** (2019). *Why are mayflies (Ephemeroptera) lost following small increases in salinity? Three conceptual osmophysiological hypotheses*. Phil. Trans. R. Soc. B 374, 20180021. (doi:10.1098/rstb. 2018.0021)

[18] **Buchwalter D, Scheibener S, Chou H, Soucek D, Elphick J.** (2019) *Are sulfate effects in the mayfly Neocloeon triangulifer driven by the cost of ion regulation?* Phil. Trans. R. Soc.

[19] **Flowers, T. J.** (2004). *Improving crop salt tolerance.* Journal of Experimental Botany, 55(396), 307-319.

[20] **Munns, R., & Tester, M.** (2008). *Mechanisms of salinity tolerance.* Annual Review of Plant Biology, 59, 651-681.

[21] **Parida, A. K., & Das, A. B.** (2005). *Salt tolerance and salinity effects on plants: A review.* Ecotoxicology and Environmental Safety, 60(3), 324-349.

[22] **Maggio, A., Hasegawa, P. M., Bressan, R. A., Consiglio, M. F., & Joly, R. J.** (2001). *Review: Unravelling the functional relationship between root anatomy and stress tolerance.* Functional Plant Biology, 28(10), 999-1004.

[23] **Munns, R., & Termaat, A.** (1986). *Whole-Plant Responses to Salinity.* Functional Plant Biology, 13(1), 143-160.

[24] **Munns, R., & Tester, M.** (2008). *Mechanisms of salinity tolerance.* Annual Review of Plant Biology, 59, 651-681.

[25] **Hasegawa, P. M., Bressan, R. A., Zhu, J. K., & Bohnert, H. J.** (2000). *Plant cellular and molecular responses to high salinity.* Annual Review of Plant Physiology and Plant Molecular Biology, 51, 463-499.

[26] **Munns, R.** (2002). *Comparative physiology of salt and water stress.* Plant, Cell & Environment, 25(2), 239-250.

[27] **Zhu, J.-K.** (2007). *Plant Salt Stress.* John Wiley & Sons, Ltd.

[28] **Cramer, G. R., & Nowak, R. S.** (1992). *Supplemental manganese improves the relative growth, net assimilation and photosynthetic rates of salt-stressed barley.* Physiologia Plantarum, 84(4), 600-605.

[29] **Larcher, W.** (1980). *Physiological plant ecology.* In 2nd totally rev. edition ed., (pp. 303). Berlin and New York: Springer-Verlag.

[30] **Catalán J., Sanchez de la Torre & Agueda J.** (1967). *Estudio de litofacies en la cuenca receptora de los ríos Guadalteba, Guadalhorce y Turón*. Instituto de Geología Económica del C.S.I.C. Informe inédito.

[31] **Pliego J.M.** (1976). *Informe sobre la salinidad del río Guadalhorce.* Centro de Estudios Hidrográficos. Informe inédito.

[32] **Carrasco F.** (1979). *Captación de manantiales salinos subacuáticos en el fondo de embalses. Manantial Meliones.* Hidrogeología y Recursos Hidráulicos 4: 465-479.

[33] **Carrasco F.** 1986. *Contribución al conocimiento de la cuenca alta del río Guadalhorce. El medio físico.* Hidrogeoquímica. Tesis Doctoral Universidad de Granada, 435 pp.

[34] **Troyano F. & Díaz J.C.** (2006). *Recuperación de las aguas salobres del río Guadalhorce por desalación*. III congreso de Ingeniería Civil. Territorio y Medio Ambiente.

[35] **A. El Bilali and A. Taleb,** (2020) *Prediction of irrigation water quality parameters using machine learning models in a semi-arid environment,* Journal of the Saudi Society of Agricultural Sciences.

[36] **Chou, Jui-Sheng & Ho, Chia-Chun & Hoang, Ha-Son.** (2018). *Determining quality of water in reservoir using machine learning.* Ecological Informatics. 44. 57–75. 10.1016/j.ecoinf.2018.01.005.

[37] **Thukral, A.; Bhardwaj, R.; Kaur, R.** (2005) *Water quality indices.* 1, 99.

[38] **Srivastava, G.; Kumar, P.** (2013) *Water quality index with missing parameters.* Int. J. Res. Eng. Technol. 2, 609–614.

[39] **Ahmed, U.; Mumtaz, R.; Anwar, H.; Shah, A.A.; Irfan, R.; García-Nieto, J.** (2019) *Efficient Water Quality Prediction Using Supervised Machine Learning.* Water 2019, 11, 2210.

[40] **Ankit Gupta, Elliott Ruebush.** (2019). *AquaSight: Automatic Water Impurity Detection Utilizing Convolutional Neural Networks.* arXiv:1907.07573

[41] **Gazzaz, N.M.; Yuso, M.K.; Aris, A.Z.; Juahir, H.; Ramli, M.F.** (2012). *Artificial neural network modeling of the water quality index for Kinta River (Malaysia) using water quality variables as predictors.* Mar. Pollut. Bull. 64, 2409–2420.

[42] **Sakizadeh, M.** (2016) *Artificial intelligence for the prediction of water quality index in groundwater systems.* Model. Earth Syst. Environ**.** 2, 8. https://doi.org/10.1007/s40808-015-0063-9

[43] **Julià Minguillón.** *Fundamentos de data science.* UOC. PID_00235534

[44] **Gu Jifa; Zhang Lingling** (2014). *«Data, DIKW, Big Data and data science».* Procedia Computer Science (vol. 31, págs. 814-821). ISSN 1877-0509. http://0-dx.doi.org.cataleg.uoc.edu/10.1016/j.procs.2014.05.332

[45] http://www.redhidrosurmedioambiente.es/saih/

[46] **Berners-Lee, T.** (2012). 5-*star deployment scheme.*http://5stardata.info/en/

[47] **Jiawei, H.; Kamber, M.; Jian, P.** (2013). *Data Mining: Concepts and Techniques (3rd).* San Francisco: Morgan Kaufmann Publishers

[48] **Brandon Butcher & Brian J. Smith** (2020) *Feature Engineering and Selection: A Practical Approach for Predictive Models,* The American Statistician, 74:3, 308-309, DOI: 10.1080/00031305.2020.1790217

[49] https://en.wikipedia.org/wiki/Linear_interpolation

[50] https://en.wikipedia.org/wiki/Polynomial_interpolation

[51] https://en.wikipedia.org/wiki/Spline_interpolation

[52] https://www.iue.tuwien.ac.at/phd/rottinger/node60.html

[53]https://www.cienciadedatos.net/documentos/py10-regresion-lineal-python.html

[53]https://towardsdatascience.com/everything-you-need-to-know-about-min-max-normalization-in-python-b79592732b79

[54] **Pedregosa *et al.*** (2011), *Scikit-learn: Machine Learning in Python,* JMLR 12, pp. 2825-2830.

[55] **L. Shanget *al.*** (2009) *Selection and optimization of cut-points for numeric attribute values.* Computers and Mathematics with Applications 57 1018–1023.

[56] **Liu H., Hussain F., Tan C., & Dash M.** (2002). *Discretization: An Enabling Technique.* Data Mining and Knowledge Discovery, 6(4), 393-423.

[57]https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.cut.html

[58] **John Hearty** (2016). *Advanced Machine Learning with Python.* Packt Publishing Ltd.

[59]https://towardsdatascience.com/feature-selection-using-python-for-classification-problem-b5f00a1c7028

[60] **Jason Brownlee** (2016). *Machine Learning Mastery with Python.* Machine Learning Mastery.

[61] **E. Alpaydin.** (2011). *Introduction to Machine Learning (2nd ed.)*. Cambridge, MA: MIT Press.

[62] **Andriy Burkov.** (2019) *The Hundred-Page Machine Learning Book.* Andriy Burkov.

[63] https://medium.com/analytics-vidhya/splitting-the-dataset-into-three-sets-78f419f0d608

[64] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

[65] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

[66] **Jason Brownlee.** (2018). *Statistical Methods for Machine Learning.* Machine Learning Mastery.

[67] **Sebastian Raschka.** (2015). *Python Machine Learning.* Packt Publishing Ltd.

[68] **Anna Bosch Rué, Jordi Casas Roma y Toni Lozano Bagén.** (2019). *Deep learning. Principios y fundamentos.* Editorial UOC.

[69] **V. N. Vapnik.** (1998). *Statistical Learning Theory*. JohnWiley & Sons.

[70]https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

[71]https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[72] https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74

[73] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[74] **Jake VanderPlas.** (2016). *Python Data Science Handbook: Tools and Techniques for Developers: Essential Tools for working with Data.* O'Reilly Media, Inc.

[75]https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html

[76] https://keras.io/

[77] Your Home for Data Science. URL, [online] Available: http://kaggle.com.

[78] **B. Pavlyshenko.** (2018). *"Using Stacking Approaches for Machine Learning Models,"* IEEE Second International Conference on Data Stream Mining & Processing (DSMP), Lviv, 2018, pp. 255-258, doi: 10.1109/DSMP.2018.8478522.

[79] https://pycaret.org/

[80] **Juan Vidal Gil.** (2012) *Gestión de datos en un Data Warehouse*. UOC Editorial.

7. Anexos

7.1 Anexo 1. Código Python

https://github.com/calonso-ai/Salinidad-de-aguas-UOC