CASO DE ESTUDIO

Identificación de patrones submarinos en especies de macroinvertebrados de monitoreo ecológico

Asignatura

Minería de Datos

Integrantes

Abderrahmane Guermat

Vanessa Lucía Ramos Rodriguez

Ana María Ortiz Legación

Kelly Turbay

Junio 2023

ÍNDICE

1. INTRODUCCIÓN

El monitoreo ecológico de la Fundación Charles Darwin (FCD) - que ha sido guardiana de y ha protegido especies como la tortuga de Galápagos, los tiburones, los pingüinos de Galápagos, entre otra fauna y flora endémicas - es un programa que brinda información sobre el estado de la biota asociada a los fondos rocosos duros y permite determinar la naturaleza y magnitud de sus fluctuaciones a lo largo del tiempo y el espacio. Este programa permite evaluar la respuesta biológica frente a los factores ambientales y antropogénicos mediante el seguimiento e investigación a largo plazo de la biodiversidad, la composición y el funcionamiento de los sistemas marinos costeros.

La pérdida de biodiversidad es una de las principales preocupaciones actuales, y que la solución a la crisis de la biodiversidad requiere la participación de múltiples grupos, desde los gobiernos y la comunidad científica hasta la sociedad civil (IPBES, 2019), sin embargo, sin información consistente y confiable sobre el estado de la biodiversidad que respalde los esfuerzos de conservación, se puede hacer poco o nada para combatir las crisis de la biodiversidad y la extinción (DPNG, 2014).

La FCD es consciente de que el hecho de contar con datos a largo plazo es imprescindible para detectar cambios, ya sean impactos acumulativos o crónicos, así mismo, entiende que comprender esta variación ecológica, así como las tendencias, es importante para desarrollar estrategias de manejo (Hewitt & Thrush, 2007). En este sentido, el Programa de Monitoreo Ecológico Submareal brinda la oportunidad no solo de observar, sino también de reaccionar ante los nuevos cambios en los ecosistemas, como los cambios de fase, la reducción de las poblaciones de peces, la invasión de especies no nativas, la disminución de especies de interés turístico y las posibles amenazas derivadas de los eventos de El Niño y el cambio climático. En consecuencia, este programa proporciona una herramienta valiosa para implementar medidas de manejo en la Reserva Marina de Galápagos.

2. OBJETIVO DEL PROYECTO

Extraer patrones de comportamiento de las especies de macroinvertebrados sobre las áreas de captura de datos (transectos), analizando variables del entorno, temporales y características de las especies. Concretamente, se han seleccionado como variables objetivo las variables que corresponden a la suma de todos los individuos contados a lo largo del transecto para una determinada especie (‘Sum_ind’) y al Status de Área Protegida Marina (‘MPA_Status’) las cuales se consideran como variables de respuesta a predecir en los modelos estudiados.

3. INFORMACIÓN SOBRE EL DATASET

De acuerdo a lo mencionado por la FCD, los datos provienen de censos visuales de macroinvertebrados marinos.

En cuanto a cómo se obtuvo la información: El monitoreo de macroinvertebrados móviles se enfoca en la medición simultánea de la densidad y abundancia de varias especies a la vez, incluyendo especies comerciales y no comerciales. Para esto, se evalúa principalmente la densidad en cada uno de los 20 cuadrantes de 1 m por 5 m ubicados a lo largo de una cinta de 50 m de longitud sobre el sustrato, todos a la misma profundidad. La identificación de especies en los cuadrantes se realiza a ambos lados de la cinta. Durante la identificación, se cuenta el número de individuos para realizar una estimación de abundancia, y también se toman medidas de tamaño según el grupo de monitoreo.

Nombre del dataset: Macroinvertebrados.xlsx

Campos que componen el dataset:

- **id:** Id representa año del monitoreo
- **dive_date:** Fecha de la inmersión de buceo
- **dive_month:** Mes de la inmersión de buceo
- **year:** Año de la inmersión de buceo
- **Transect.code:** Código de transecto/buceo
- **Island:** Isla dónde se tomó la muestra
- **Bioregion:** Bioregion dónde se tomó la muestra
- **MPA_Status:** Status de Área Protegida Marina (APM)
- **Sum_ind:** Suma de todos los individuos contados a lo largo del transecto
- **Countsize_ind:** Número de individuos de esa especie medidos en el transecto

- **TaxonID:** ID taxonomico de la colección FCD
- **Domain:** Dominio taxonomico de la colección FCD
- **Kingdom:** Reino taxonomico de la colección FCD
- **PhylumOrDivision:** Filo o división taxonomico de la colección FCD
- **Class:** Clase taxonomico de la colección FCD
- **Order:** Orden taxonomico de la colección FCD
- **Family:** Familia taxonomico de la colección FCD
- **ScientificName:** Nombre científico de la especie
- **CommonNameEnglish:** Nombre común en Inglés
- **CommonNameSpanish:** Nombre común en Español
- **Site:** Lugar del monitoreo
- **Latitude:** Latitud del monitoreo
- **Longitude:** Longitud del monitoreo
- **Subzone.name:** Categorías de zonificación
- **Refuge_Level:** Nivel de refugio
- **depth_strata:** Estratos de profundidad
- **epoca:** Epoca del año

4. DESCRIPCIÓN DE LOS DATOS

Para analizar datos, es necesario tener una idea general de las características de un conjunto de datos. Para ello, se utilizarán diversas herramientas que nos permitirán conocer los datos a tratar y analizar los mismos.

Composición de los datos

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7025 entries, 0 to 7024
Data columns (total 27 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 7025 non-null   int64
 1   dive_date          6933 non-null   datetime64[ns]
 2   dive_month         6933 non-null   object
 3   year               7025 non-null   int64
 4   Transect.code      7025 non-null   object
 5   Island             6919 non-null   object
 6   Bioregion          6980 non-null   object
 7   MPA_Status         7025 non-null   object
 8   Sum_ind            7025 non-null   int64
 9   Countsize_ind      7025 non-null   int64
 10  TaxonID            6782 non-null   float64
 11  Domain             6782 non-null   object
 12  Kingdom            6782 non-null   object
 13  PhylumOrDivision   6782 non-null   object
 14  Class              6782 non-null   object
 15  Order              6782 non-null   object
 16  Family             6782 non-null   object
 17  ScientificName     7025 non-null   object
 18  CommonNameEnglish  6687 non-null   object
 19  CommonNameSpanish  6681 non-null   object
 20  Site               7025 non-null   object
 21  Latitude           6980 non-null   float64
 22  Longitude          6980 non-null   float64
 23  Subzone.name       6980 non-null   object
 24  Refuge_Level       6937 non-null   object
 25  depth_strata       7025 non-null   object
 26  epoca              7025 non-null   object
dtypes: datetime64[ns](1), float64(3), int64(4), object(19)
memory usage: 1.4+ MB
```

Figura 1: Composición del dataset

En la figura 1, podemos observar que el dataset está compuesto por 7.025 filas y 27 columnas. Cada fila representa un registro que contiene la información de una especie determinada y cada columna corresponde a una característica de la muestra.

Así mismo, se observa que el dataset está compuesto por:

- 19 campos tipo object que pueden corresponder a tipo texto o una combinación de datos de texto y otros tipos de datos.
- 4 campos de tipo int64.
- 3 campos de tipo float64.
- 1 campo de tipo datetime64

La memoria utilizada por el objeto es de 1.4+ MB.

- **Información estadística**

	Sum_ind	Countsize_ind
count	7025.000000	7025.000000
mean	60.198719	4.520712
std	156.153490	15.697667
min	0.000000	0.000000
25%	1.000000	0.000000
50%	3.000000	1.000000
75%	22.000000	3.000000
max	2125.000000	423.000000

Figura 2: Información estadística

En la figura 2, se muestra un resumen estadístico de un dos de las columnas numéricas del dataset, incluyendo medidas como el número de observaciones, la media, la desviación estándar, la mediana, el valor mínimo y máximo. Es importante mencionar que si hay valores faltantes (NaN) en los datos, el método los ignorará en los cálculos realizados. En este caso, no se puede concluir algo con esta información, dado que la data presenta gran variedad. Es decir, que corresponde a distintos años, ubicación y especies, etc. De todas maneras, nos permite ver la información de manera general.

- **Valores únicos**

	Columna	Valores únicos
0	Island	[Española, Fernandina, Floreana, Isabela, Sant...
1	Bioregion	[Sureste, Oeste, Bahía Elizabeth, Lejano Norte...
2	MPA_Status	[Extractive use, Sanctuary]
3	Domain	[Eukaryota, nan]
4	Kingdom	[Animalia, nan, Chromalveolata (= Chromista)]
5	PhylumOrDivision	[Echinodermata, Mollusca, nan, Annelida, Arthr...
6	Class	[Echinoidea, Holothuroidea, Gastropoda, nan, A...
7	Order	[Diadematoida, Cidaroida, Holothuriida, Camaro...
8	Family	[Diadematidae, Cidaridae, Holothuriidae, Toxop...
9	ScientificName	[Diadema mexicanum, Eucidaris galapagensis, Ho...
10	CommonNameEnglish	[Hatpin urchin, Slate pencil urchin, Sea cucum...
11	CommonNameSpanish	[erizo aguja, erizo lapicero, pepino de mar, e...
12	Subzone_name	[Conservación, Intangible, Aprovechamiento Sus...
13	Site	[ES01-Bahía Gardner Norte (1), ES02-Cerro Colo...
14	Refuge_Level	[Extractive use, Sanctuary, nan]
15	depth_strata	[6m, 15m, 12m, 10m, 11m, -]
16	epoca	[Caliente, Fría]

Figura 3. Valores únicos

La figura 3 permite ver una muestra de los valores únicos que componen las columnas del dataset, e identificar valores nulos o inconsistentes en la información (como en el caso de las columnas Island, Bioregion, Domain, Kingdom, PhylumOrDivision, Subzone_name, Refuge_Level), los cuales deberán ser limpiados el apartado de preprocesamiento de la información.

	Columna	Valores Nulos
0	dive_date	92
1	dive_month	92
2	year	0
3	TransectCode	0
4	Island	106
5	Bioregion	45
6	Domain	243
7	Kingdom	243
8	PhylumOrDivision	243
9	Subzone_name	45
10	Refuge_Level	88
11	Sum_ind	0
12	Countsize_ind	0
13	TaxonID	243
14	Latitude	45
15	Longitude	45

Figura 4. Valores nulos

En la figura 4, se observa que:

- Para las columnas Domain, Kingdom, PhylumOrDivision y TaxonID existen 243 valores nulos, lo cual representa un 3.46% del total de la data.
- Para la columna Island existen 106 valores nulos, lo cual representa un 1.5% del total de la data.
- Para la columna dive_date y dive_month se observa que existen 92 valores nulos, lo cual representa un 1.09% del total de la data.
- Para la columna Refuge_Level se observa que existen 88 valores nulos, lo cual representa un 1.25% del total de la data.
- Para la columna Bioregion, Subzone_name, Latitude y Longitude se observa que existen 45 valores nulos, lo cual representa un 0.64% del total de la data.

- **Valores únicos**

```
id                    10
dive_date            135
dive_month             8
year                  10
TransectCode        1093
Island                 9
Bioregion              5
MPA_Status             2
Sum_ind              606
Countsize_ind        109
TaxonID               76
Domain                 1
Kingdom                2
PhylumOrDivision       6
Class                 11
Order                 22
Family                47
ScientificName        90
CommonNameEnglish     61
CommonNameSpanish     61
Site                 100
Latitude              97
Longitude             97
Subzone_name           3
Refuge_Level           2
depth_strata           6
epoca                  2
dtype: int64
```

Figura 5. Valores únicos

La figura 5 nos permite observar qué columnas cuentan con un solo valor. Esta información será de utilidad al momento de seleccionar las características que serán considerados para el modelo.

- **Análisis de los datos**

En este apartado, se responderá a los distintos tipos de cuestionamientos con respecto al dataset elegido.

A continuación, las preguntas planteadas:

¿Cómo ha sido la distribución de las especies según la isla (Island) donde se encuentran?

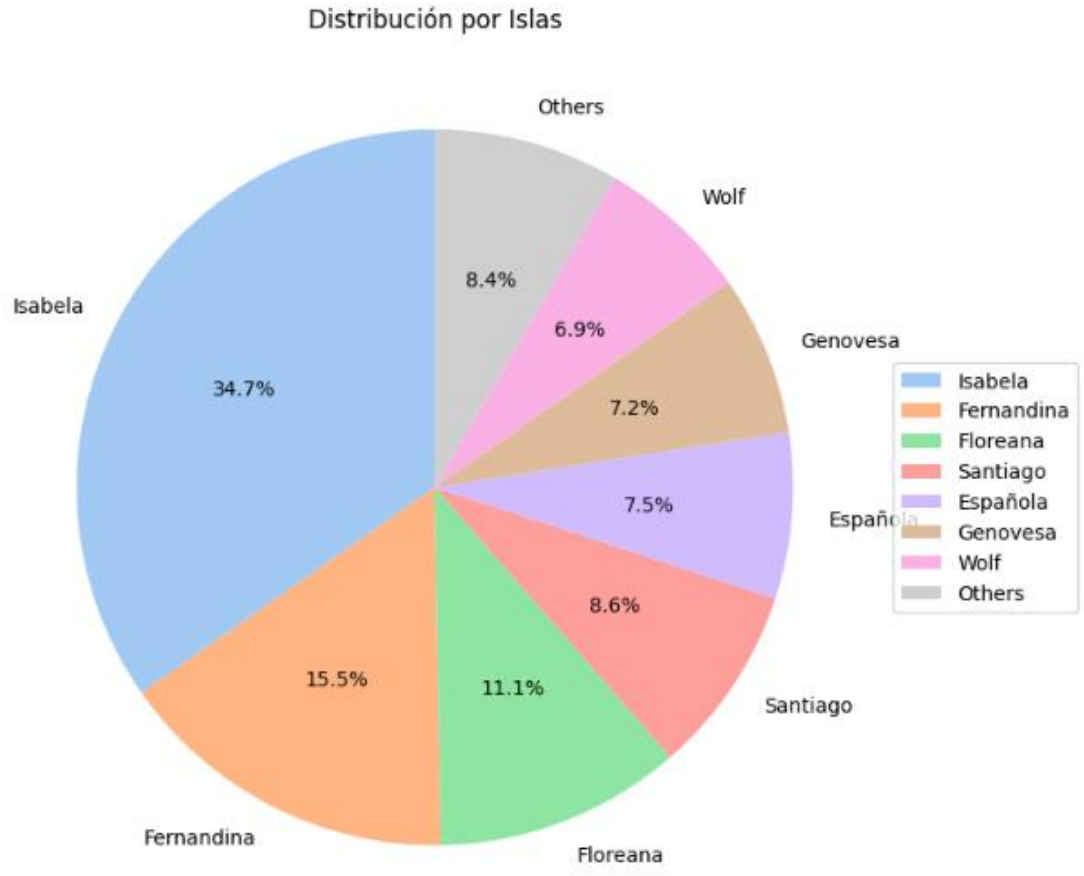

Figura 6. Distribución por islas

Como se puede observar en la Figura 6, la mayor cantidad de registros presentes en el dataset corresponden a información de la isla Isabela (34,7%), seguido por la isla Fernandina (15.5%) y Floreana (11.1%).

¿Cuales son los Reinos Animales (Kingdom) presentes en el dataset y qué Filos o Divisiones (PhylumOrDivision) lo componen?

El dataset esrá compuesto por dos reinos (kingdoms): Animalia y Chromalveolata, del dominio (domain) Eukaryota. Sin embargo, no hay especies contabilizadas en el reino Chromalveolata, por lo que en el siguiente gráfico solo se muestra la distribución del reino Animalia.

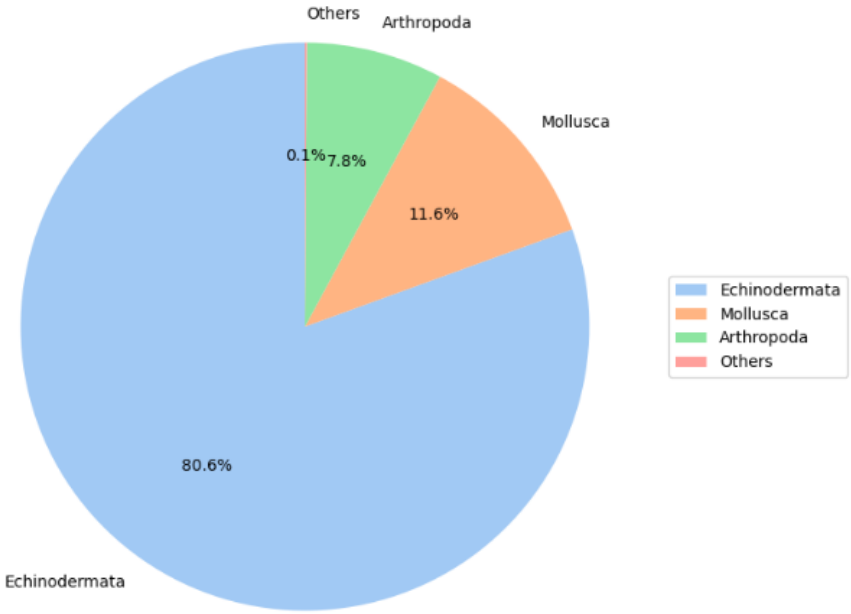

Figura 7. Distribución por Filos

Como se observa en la Figura 7, el datset está compuesto principalmente por tres Filos o Divisiones del reino animal: Echinodermata (con un 80.6%), Mollusca (con 11.6%) y Arthropoda (con 7.8%).

¿Cuáles son las Clases (Class) del Reino animal presentes en este dataset y como están distribuidas según sus Órdenes (Order)?

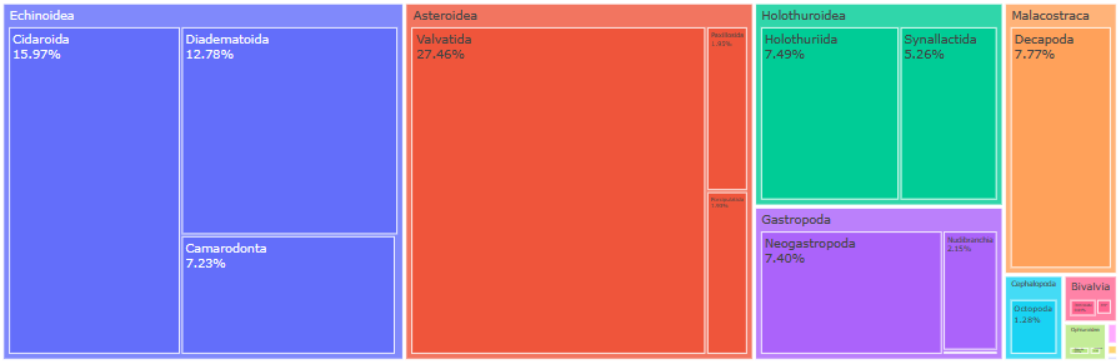

Figura 8. Distribución por Clases y Órdenes

Como se puede observar en la Figura 8, cada color corresponde a una clase dentro del dataset. Así mismo, dentro de cada clase también se pueden observar las órdenes que la componen, y su distribución porcentual dentro del datset. Por ejemplo, podemos observar

que la orden "Valvatida" es la que contiene la mayor cantidad de registros dentro del dataset, con un 27.46% del total de los registros, seguido por la orden "Cidaroida", con 15.97%.

¿Cuáles son las Órdenes (Order) del reino animal presentes en este dataset y como está distribuido según sus Familias (Family)?

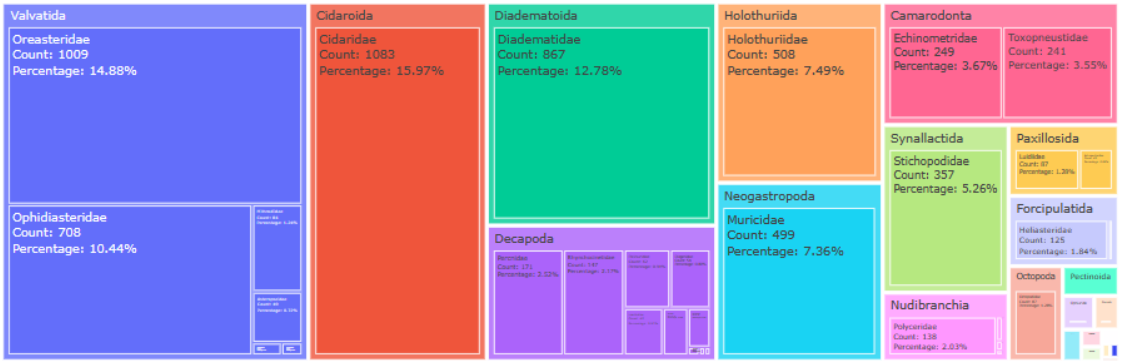

Figura 9. Distribución por Órdenes y Familias

En la Figura 9, cada color corresponde a una Orden dentro del dataset. Así mismo, dentro de cada caja que corresponde al Orden también se pueden observar las familias que la componen y su distribución porcentual dentro del datset. Por ejemplo, podemos observar que la orden "Cidaridae" es la que contiene la mayor cantidad de registros dentro del dataset, con un 15.97% del total de los registros, seguido por la orden "Valvatida", con 14.88%.

¿Cuántos sitios (Sites) componen cada isla (Island) y como está distribuido en el dataset?

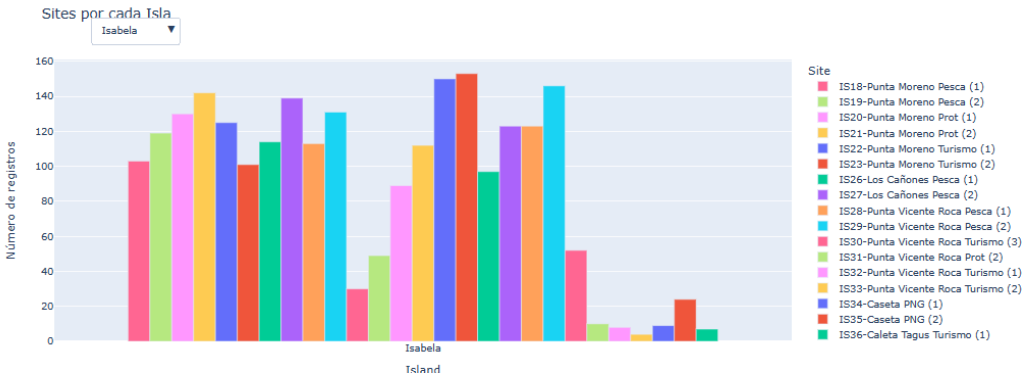

Figura 10. Sites por Isla

La figura 10 permite ver los sitios (Sites) que componen cada isla (Island), y de qué manera están distribuidos dentro del dataset. Este es un gráfico dinámico al que se puede acceder desde el Notebook que acompaña este informe.

¿Cómo está distribuído el dataset según su Subzona (subzone.name), Nivel de Refugio (Refuge_Level), Estrato de profundidad (depth_strata) y época?

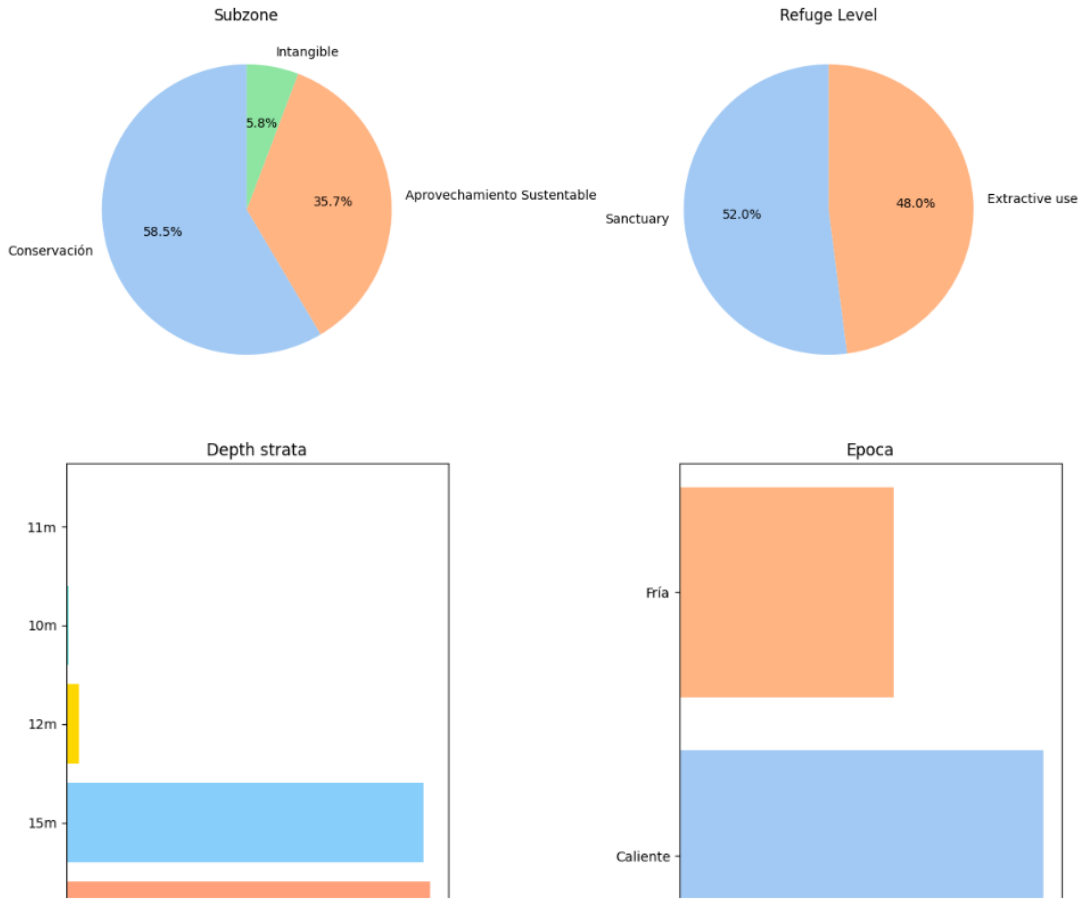

Figura 11. Distribución por Subzone, Refuge Level, Depth Strata y Época

Tal como se puede ver en la Figura 11, se puede apreciar que, en cuanto a la subzona (Subzone) se refiere, el dataset tiene una mayor cantidad de registros para la subzona "Conservación", con 58.5%, seguida por "Aprovechamiento Sustentable", con 35.7%. Por su parte, la subzona "Intangible" solo representa el 5.8% del total de los registros del dataset.

En cuanto al Nivel de refugio (Refuge level), "Sanctuary" tiene el 52% de los registros mientras que, por su parte, "Extrative use" solo tiene el 48% del total de los registros.
En cuanto al Estrato de profundidad (depth_strata), se observa que es más común encontrar registros con estrato de profundidad de 6 o 15 metros, siendo menos probable que se encuentren registros con 10, 11 o 12 metros de estrato de profundidad.
En cuanto a la época, podemos observar que existe una mayor cantidad de registros que contienen valor "Caliente" para este campo, sobre la cantidad de registros que contienen como valor "Frío".

5. PREPROCESAMIENTO DE LOS DATOS

El objetivo del preprocesamiento de datos es limpiar, transformar y preparar los datos para su análisis. En este apartado se han considerado técnicas de limpieza de datos, normalización de los datos, eliminación de valores atípicos, selección de características relevantes y la creación de nuevas características a partir de los datos existentes, selección por varianza, selección por valores ausentes, selección mediante visualización, entre otros.

- **Ajustes para valores ausentes o anómalos**

  Se eliminaron las filas que contienen valores nulos en campos importantes para el modelo, pero que no pueden ser imputadas.

  A continuación, se listan las acciones realizadas en relación a este apartado.

  - Algunas especies como "Holothuria (Stauropora) fuscocinerea", "Triplofusus princeps", etc, presenta valores nulos en las columnas que corresponden a su clasificación en el reino animal (Domain, Kingdom, PhylumOrDivision, Class, Order, Family). Dado que consideramos que estos valores son importantes para nuestro modelo, retiraremos a estas especies del dataset.
  - Dado que "Bioregion" es un campo importante para el modelo y no puede ser imputado con la información que se encuentra en el dataset, los registros que contenían valores nulos para este campo fueron eliminados.
  - Se detectó que existen 91 registros que tienen valores nulos para el campo "dive_date" y "dive_month". Dado que es un campo importante para el modelo, y no puede ser imputado con la información que se encuentra en el dataset, los registros que contienen valores nulos para este campo fueron eliminados.

```
id                    0
dive_date             0
dive_month            0
year                  0
TransectCode          0
Island              106
Bioregion             0
MPA_Status            0
Sum_ind               0
Countsize_ind         0
TaxonID               0
Domain                0
Kingdom               0
PhylumOrDivision      0
Class                 0
Order                 0
Family                0
ScientificName        0
CommonNameEnglish    92
CommonNameSpanish    97
Site                  0
Latitude              0
Longitude             0
Subzone_name          0
Refuge_Level         41
depth_strata          0
epoca                 0
dtype: int64
```

Figura 11. Valores nulos tras la primera parte del pre-procesamiento

- **Imputación de valores ausentes**

  Otra de las columnas que consideramos importante es la Isla (Island). Como se observa, ahora existen 106 registros que tienen valores nulos en este campo. Al analizar la informaciòn, también se observa que todos estos registros tienen en el campo Transecto un valor que empieza por PI, por lo que si se averigua este valor, se podría completar la información. Sin embargo, por lo pronto, se decidió colocar un valor para poder identificarlo, llamado "PI-PorIdentificar".

- **Eliminación de columnas no relevantes para el modelo**

  Dado que los valores de "Refuge_Level" son iguales a los de "MPA_Status" se eliminará esta columna. Así mismo, dado que los nombres de las especies en inglés y en español (CommonNameEnglish y CommonNameSpanish) no son importantes para el modelo (ya que tenemos el nombre científico de la especie) se eliminarán estas columnas del dataset.

```
id                  0
dive_date           0
dive_month          0
year                0
TransectCode        0
Island              0
Bioregion           0
MPA_Status          0
Sum_ind             0
Countsize_ind       0
TaxonID             0
Domain              0
Kingdom             0
PhylumOrDivision    0
Class               0
Order               0
Family              0
ScientificName      0
Site                0
Latitude            0
Longitude           0
Subzone_name        0
depth_strata        0
epoca               0
dtype: int64
```

Figura 12. Valores nulos tras la segunda parte del pre-procesamiento

- **Transformaciones y selección de características**

Técnica One-hot:

La técnica "one-hot", también conocida como codificación "one-hot encoding", es una forma común de representar variables categóricas en forma numérica en el aprendizaje automático. Convierte cada categoría en una columna binaria y permite capturar la información categórica de manera adecuada para su uso en algoritmos de aprendizaje automático.

En el aprendizaje automático, muchos algoritmos requieren que los datos de entrada estén en forma numérica para poder realizar cálculos y modelar relaciones matemáticas. Sin embargo, las variables categóricas, como algunas de las variables que contiene el dataset con el que venimos trabajando, no se pueden utilizar directamente en su forma original en la mayoría de los algoritmos de aprendizaje automático.

La codificación "one-hot" resuelve este problema al crear una representación binaria para cada categoría única en una variable categórica. El proceso implica convertir cada categoría en una nueva columna y asignar un valor de 1 en la columna correspondiente a la categoría presente, y un valor de 0 en todas las demás columnas.

La ventaja de la codificación "one-hot" es que permite capturar la información categórica de manera adecuada sin asignar ningún orden o relación numérica incorrecta entre las categorías. Además, al representar cada categoría en una columna separada, evita el problema de asignar importancia relativa incorrecta a las categorías.

Cabe mencionar también que la codificación "one-hot" puede llevar a un aumento en la dimensionalidad de los datos, especialmente cuando se tienen muchas categorías únicas en una variable.

En este punto, se utilizó la técnica one-shot para normalizar las características del modelo, pero antes, haremos algunas otras transformaciones previas, como por ejemplo:

- Se transformaron los valores de la columna "depth_strata" a un valor numérico retirándole la m (de metros).
- Se transformó el valor de "depth_strata (m)" a valor float.
- Finalmente, se utilizó la técnica one-shot para normalizar las características del modelo.

Técnica Label-encoding:

Esta segunda técnica sirve para convertir las categorías de una variable categórica en números enteros. Es especialmente útil cuando estamos tratando con datos que contienen texto, pero nuestros algoritmos de aprendizaje automático necesitan números para trabajar eficientemente. Dada una variable a cada categoría de esta se le asigna un valor numérico diferente.

- **Dataset final (con técnica one-hot)**

A continuación, una muestra del dataset procesado con la técnica one-hot.

dive_month_April	dive_month_August	dive_month_February	dive_month_July	dive_month_June	dive_month_March	dive_month_May	dive_month_November	Island_Darwin	Island_Fernandina	...	Subzone_name_Aprovech Sust
0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	

Figura 13. Muestra del dataset one-hot

- **Dataset final (con técnica label-encoding)**

A continuación, una muestra del dataset procesado con la técnica label-encoding.

	year	month	day	Sum_ind	Latitude	Longitude	depth_strata (m)	Island_encoded	Bioregion_encoded	Site_encoded	MPA_Status_encoded	PhylumOrDivision_encoded	Class_encoded	Order_encoded
0	2010	2.0	4.0	1	-1.34421	-89.66820	6.0	0	0	0	0	0	0	0
1	2010	2.0	4.0	203	-1.34421	-89.66820	6.0	0	0	0	0	0	0	1
2	2010	2.0	4.0	4	-1.34421	-89.66820	6.0	0	0	0	0	0	1	2
3	2010	2.0	4.0	45	-1.34421	-89.66820	6.0	0	0	0	0	0	0	3
4	2010	2.0	4.0	5	-1.34421	-89.66820	15.0	0	0	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
7020	2020	11.0	19.0	362	-0.28115	-90.56861	15.0	4	0	79	1	0	0	1
7021	2020	11.0	19.0	1	-0.28115	-90.56861	15.0	4	0	79	1	-1	-1	-1
7022	2020	11.0	19.0	1	-0.28115	-90.56861	15.0	4	0	79	1	0	3	5
7023	2020	11.0	19.0	79	-0.28115	-90.56861	15.0	4	0	79	1	0	3	5
7024	2020	11.0	19.0	1	-0.28115	-90.56861	15.0	4	0	79	1	0	0	3

6880 rows × 18 columns

Figura 14. Muestra del dataset label-encoding

6. MODELOS DE PREDICCIÓN

5.1. TAREA DE REGRESIÓN

En esta tarea, se ha seleccionado como variable a predecir la correspondiente a 'Sum_ind', la cual indica la suma de todos los individuos contados a lo largo del transecto. Así, el objetivo de este apartado es conseguir el modelo óptimo para predecir el número total de individuos que pueden llegar a ser encontrados en una inmersión de buceo, en base a unas determinadas características, como pueden ser la ubicación o la época del año, entre otros.

Concretamente, se emplearán diferentes técnicas de regresión con el fin de obtener el mejor modelo que pueda predecir la variable 'Sum_ind' a partir de las características y de las variables existentes en el conjunto de datos. Cabe destacar que durante este estudio se utiliza el dataset final que ha sido trabajado mediante la técnica one-hot.

En el siguiente apartado se muestran los algoritmos utilizados, además de un análisis de los resultados obtenidos.

5.1.1. DecisionTreeRegressor

El primer algoritmo estudiado corresponde al algoritmo DecisionTreeRegressor, el cual se implementará con diferentes parámetros con el fin de observar su efecto.

Concretamente, el algoritmo DecisionTreeRegressor toma como parámetro el nivel de profundidad del árbol ('max_depth'). Así, a la hora de implementar el algoritmo se han

tomado diferentes valores para ‘max_depth’, siendo estos valores comprendidos entre 1 y 20. Por otro lado, cabe destacar que el modelo ha sido evaluado mediante R2.

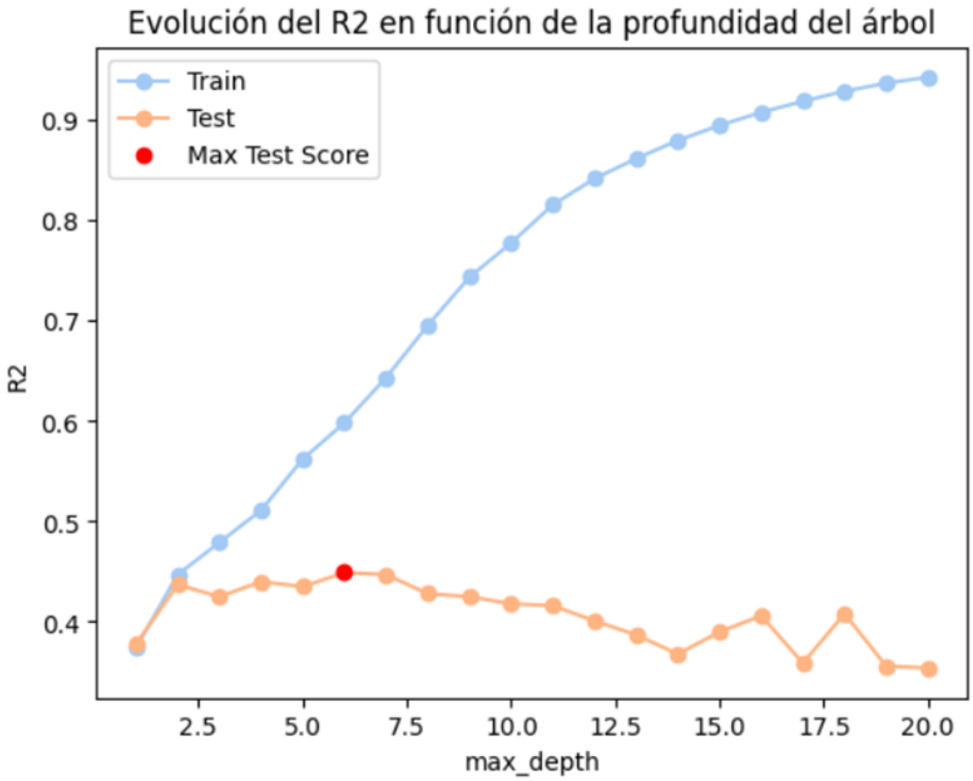

Figura 15. Evolución del rendimiento del algoritmo DecisionTreeRegressor con diferentes valores de ‘max_depth’

En la gráfica anterior se muestra la evolución del valor R2 en función del parámetro del nivel de profundidad del árbol o ‘max_depth’. Concretamente, en azul claro se representan los valores de R2 para el conjunto de datos de entrenamiento, mientras que en naranja claro se observa el valor de R2 para el conjunto de datos de test. Asimismo, se representa en color rojo el punto máximo de R2 para el conjunto de datos de test.

A partir de la gráfica se puede concluir que a medida que aumenta el valor de ‘max_depth’, el valor de R2 en el conjunto de entrenamiento aumenta acercándose a 1. No obstante, el valor de R2 en el conjunto de test obtiene su máximo valor para ‘max_depth’ igual a 6, observándose cómo el valor de R2 disminuye a partir de este punto.

Con ello, se obtiene que el modelo que mejor se ajusta a los datos para el algoritmo DecisionTreeRegressor se alcanza con el valor 'max_depth' igual a 6, siendo el valor de R2 en el conjunto de entrenamiento de 0.598 y el valor de R2 en el conjunto de test de 0.449.

Cabe destacar que, en la implementación anterior del modelo, este se evaluó utilizando una única partición de datos (70% de los datos para el entrenamiento y el 30% de los datos para el test), lo cual podría llevar a una evaluación poco fiable debido a la posible variabilidad de datos. Concretamente, en entornos reales resulta más eficiente realizar validación cruzada (cross validation). Por tanto, ahora se implementa la validación cruzada con el fin de obtener una evaluación más fiable y robusta del modelo con el algoritmo DecisionTreeRegressor.

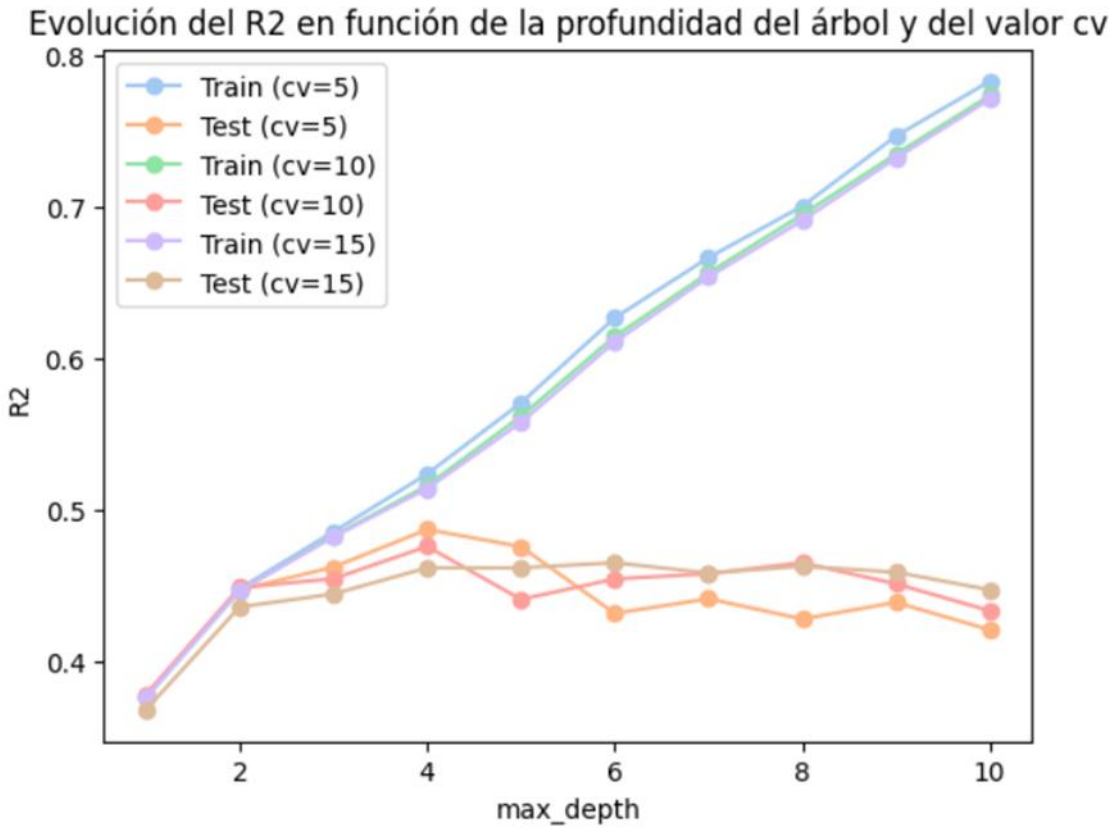

Figura 16. Evolución del rendimiento del algoritmo DecisionTreeRegressor con diferentes valores de 'max_depth' y 'cv'

En este caso, el rango de valores para la profundidad del árbol se ha reducido de 20 a 10, ya que anteriormente se ha visualizado que la métrica R2 comienza a disminuir a partir de 'max_depth' igual a 10 aproximadamente.

En la gráfica anterior, se muestran de nuevo la evolución del valor R2 en función del parámetro 'max_depth', pero añadiendo además el parámetro cv de la validación cruzada. En este sentido, se puede observar que al implementar la validación cruzada, los valores de R2 para los diferentes valores de cv son consistentes y no existe una gran variación en los resultados.

Finalmente, se tiene que el mejor resultado de R2 se alcanza para cv igual a 5 y 'max_depth' igual a 4 (el cual difiere del valor obtenido previamente sin validación cruzada), con un valor de R2 en el conjunto de test de 0.487, siendo este ligeramente mayor al 0.451 obtenido anteriormente sin validación cruzada.

5.1.2. Ridge Regression

En segundo lugar, se utiliza el algoritmo Ridge Regression implementando desde un principio la validación cruzada. El algoritmo Ridge Regression algoritmo recibe como parámetro 'alpha', por lo que se han probado distintos valores con el fin de obtener el parámetro que obtiene el mejor modelo. Concretamente, se han probado los valores [0.01, 0.1, 1, 10, 100].

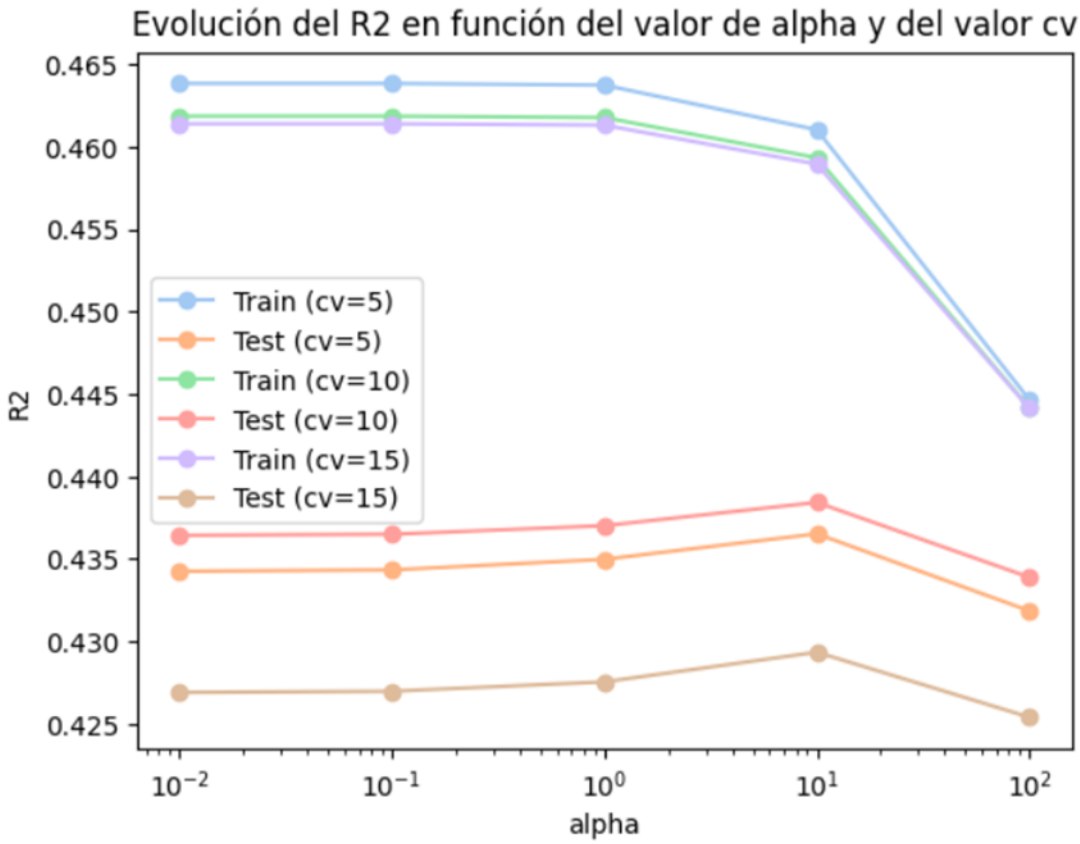

Figura 17. Evolución del rendimiento del algoritmo Ridge Regression con diferentes valores de 'max_depth' y 'cv'

Una vez observados los resultados de R2 para el algoritmo Ridge regression en función de distintos valores tanto para 'cv' como para 'alpha' se puede concluir que, en general, los valores de R2 en el conjunto de prueba son similares para los diferentes valores de 'alpha' y 'cv'. Es decir, el modelo no es muy sensible a estos parámetros en este rango de valores. No obstante, el valor óptimo se obtiene para R2 en el conjunto de datos de test igual a 0.438 para los valores de cv igual a 10 y Alpha igual a 10.

5.1.3. KNN Regression

El tercer algoritmo utilizado corresponde al algoritmo KNN Regression. Concretamente, este algoritmo debe recibir el parámetro 'n_neighbors', es por ello que se prueban

diferentes valores: [1, 3, 5, 7, 9, 11, 13, 15]. Asimismo, se vuelve a implementar la validación cruzada.

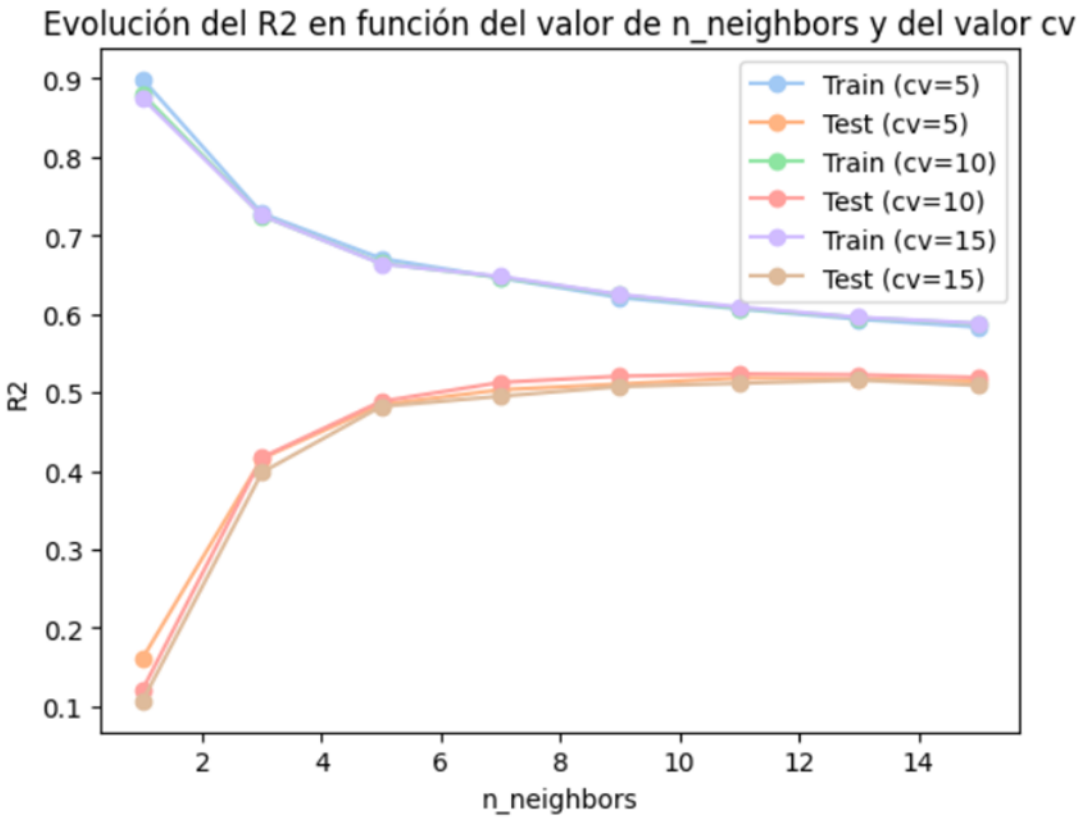

Figura 18. Evolución del rendimiento del algoritmo KNN Regression con diferentes valores de 'n_neighbors' y 'cv'

En este sentido, se observan los diferentes resultados de R2 para el algoritmo KNN Regression en función de distintos valores tanto para 'cv' como para 'n_neighbors'.

Con ello, se puede concluir que, en general, los valores de R2 en el conjunto de test son similares para los diferentes valores de 'cv'. Sin embargo, sí que se observa una diferencia en los resultados de R2 según el valor de 'n_neighbors', de manera que para valores menores a 7, la métrica R2 es considerablemente menor. Así, el valor óptimo se obtiene para R2 igual a 0.524 para los valores de cv igual a 10 y 'n_neighbors' igual a 11.

5.1.4. RandomForestRegressor

En este apartado se implementa como último algoritmo RandomForestRegressor. Concretamente, este algoritmo toma como parámetros el número de árboles o 'n_estimators', por lo que durante la implementación de dicho algoritmo se han probado diferentes valores de 'n_estimators': [10, 50, 100, 200, 300].

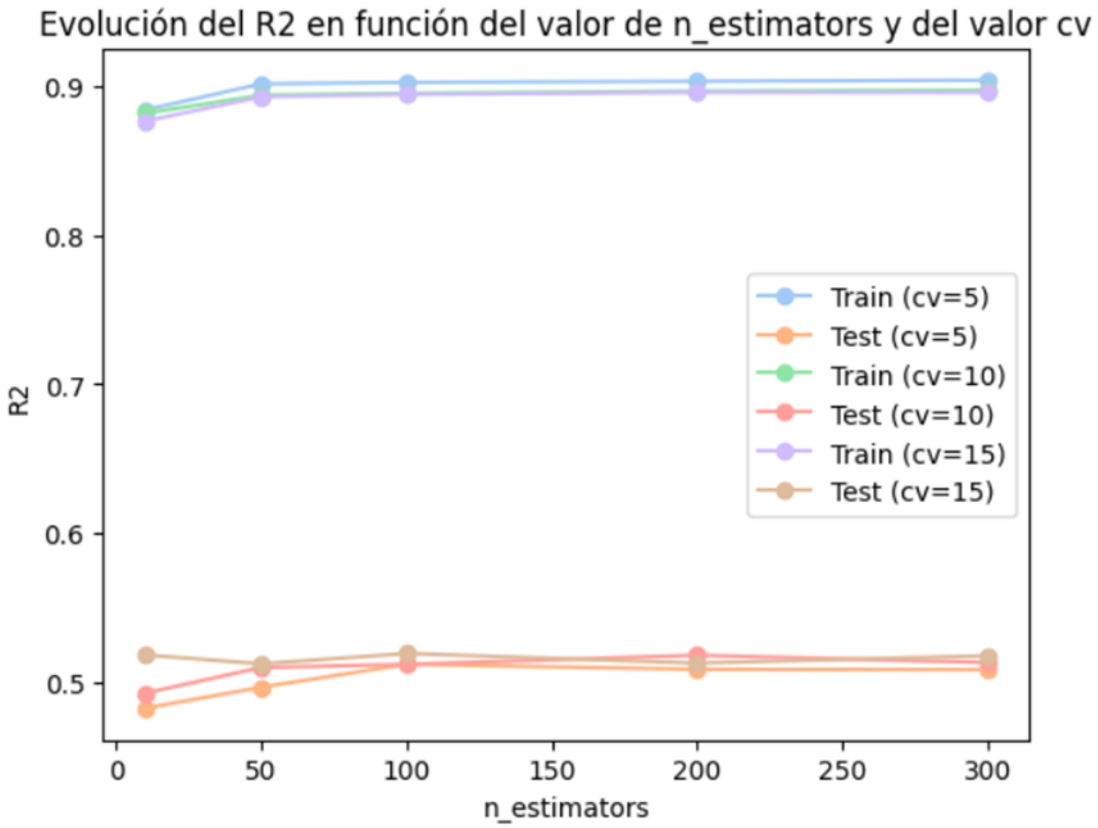

Figura 19. Evolución del rendimiento del algoritmo RandomForestRegressor con diferentes valores de 'n_estimators' y ‘cv’

Una vez obtenidos los resultados del algoritmo RandomForestRegressor, se observa que el modelo mejora a medida que aumenta el valor de 'n_estimators'. No obstante, se aprecia una tendencia de estabilización del valor de R2 cuando 'n_estimators' alcanza el valor de 100.

Con todo ello, tras la comprobación de varios valores en los parámetros se obtiene que el máximo R2 en el conjunto de test es igual a 0.519 con cv igual a 15 y n_estimators igual a 100.

5.1.4. Conclusiones

Finalmente, una vez se han implementado los modelos a comparar, se recopila la siguiente tabla:

Algoritmos	Máximo R2 (test)	R2 (train)	k (cross validation)	Otros parámetros
DecisionTreeRegressor	0.487	0.524	5	max_depth = 4
Ridge Regression	0.438	0.459	10	Alpha= 10
KNN Regressor	0.524	0.608	10	n_neighbors = 11
RandomForesRegressor	0.519	0.895	15	n estimators = 100

Tabla 1. Comparativa de los algoritmos de regresión con sus valores R2 y parámetros

La tabla anterior presenta los algoritmos de regresión implementados en el presente estudio junto con su valor R2 máximo en el conjunto de test, su valor R2 de entrenamiento correspondiente y sus parámetros correspondientes.

Con todo ello, se puede observar que el modelo obtenido a partir del algoritmo KNN Regressor es el que obtiene una métrica R2 en el conjunto de test mayor en comparación con los otros dos modelos. No obstante, es importante destacar que los resultados de los modelos implementados son relativamente bajos, lo que puede indicar que los modelos no han conseguido lograr una explicación completa de la varianza de los datos.

5.2 Tarea de regresión (Label-encoding)

En esta tarea tal y como se ha hecho en la anterior se ha seleccionado como variable a predecir la correspondiente a 'Sum_ind', la cual indica la suma de todos los individuos contados a lo largo del transecto. Así, el objetivo de este apartado sería el mismo que en el anterior. Solamente que en este caso se aplica una técnica diferente de codificación de valores textuales. Finalmente se emplearán diferentes técnicas de regresión con el fin de obtener el mejor modelo que pueda predecir la variable 'Sum_ind' a partir de las características y de las variables existentes en el conjunto de datos.

En el siguiente apartado se muestran los algoritmos utilizados, además de un análisis de los resultados obtenidos.

5.2.1 Decision Tree

Tal y como en el apartado anterior se ha aplicado el algoritmo decision Tree y a continuación se observan los resultados obtenidos:

	Feature Set	Metric	Score
0	V. Todas	r2 Train (CV 1-5:)	[0.48, 0.48, 0.47, 0.52, 0.49]
1	V. Todas	r2 Test (CV 1-5:)	[0.5, 0.49, 0.54, 0.37, 0.47]
2	V. Todas	neg_mean_absolute_error Train (CV 1-5:)	[-43.11, -42.11, -44.34, -40.91, -43.75]
3	V. Todas	neg_mean_absolute_error Test (CV 1-5:)	[-41.89, -47.15, -40.57, -48.85, -39.73]
4	V. biologicas y de region	r2 Train (CV 1-5:)	[0.44, 0.45, 0.44, 0.49, 0.45]
5	V. biologicas y de region	r2 Test (CV 1-5:)	[0.49, 0.47, 0.52, 0.36, 0.44]
6	V. biologicas y de region	neg_mean_absolute_error Train (CV 1-5:)	[-45.17, -43.87, -46.23, -42.57, -45.99]
7	V. biologicas y de region	neg_mean_absolute_error Test (CV 1-5:)	[-43.0, -48.27, -42.09, -50.55, -41.23]
8	V. temporales y de ubicacion	r2 Train (CV 1-5:)	[0.03, 0.03, 0.04, 0.03, 0.03]
9	V. temporales y de ubicacion	r2 Test (CV 1-5:)	[-0.01, -0.04, -0.0, 0.01, -0.02]
10	V. temporales y de ubicacion	neg_mean_absolute_error Train (CV 1-5:)	[-87.2, -85.3, -87.97, -84.21, -90.2]
11	V. temporales y de ubicacion	neg_mean_absolute_error Test (CV 1-5:)	[-88.67, -93.72, -82.63, -93.66, -82.91]

Figura 20. Resultado DT

En base a los resultados se puede observar lo siguiente en base al subconjunto seleccionado y a la métrica R2:

- V. Todas: En el caso de seleccionar todas las variables se observa que los valores de r2 en el entrenamiento se encuentran en el siguiente rango 0.47-0.52 y entre 0.37 y 0.54. esto significa que el algoritmo no está aprendiendo ni en el conjunto de entrenamiento ni en el de prueba
- V. Biológicas y de región: En este caso se observa que los valores de r2 en el entrenamiento se encuentran en el siguiente rango 0.44 - 0.49 y entre 0.36 y 0.52 en test. esto significa que el algoritmo no está aprendiendo ni en el conjunto de entrenamiento ni en el de prueba

- V. temporales y de ubicación: En este caso se observa que los valores de r2 en el entrenamiento se encuentran en el siguiente rango 0.03 - 0.04 y con valores similares en test. Esto sugiere que este conjunto de variables tiene una capacidad nula de predicción.

En Conclusión, respecto a la métrica R2, el conjunto de variables que ofrece los mejores resultados es "V. Todas". Sin embargo, cabe destacar que los valores son muy bajos por lo que este modelo no sería óptimo para nuestro problema.

5.2.2 Random Forest

A continuación se muestran los resultados a la hora de aplicar este algoritmo a nuestro problema:

	Feature Set	Metric	Score
0	V. Todas	r2 Train (CV 1-5:)	[0.94, 0.94, 0.94, 0.94, 0.94]
1	V. Todas	r2 Test (CV 1-5:)	[0.59, 0.56, 0.65, 0.52, 0.52]
2	V. Todas	neg_mean_absolute_error Train (CV 1-5:)	[-13.76, -13.58, -14.29, -13.14, -13.85]
3	V. Todas	neg_mean_absolute_error Test (CV 1-5:)	[-36.0, -41.06, -32.27, -42.32, -35.41]
4	V. biologicas y de region	r2 Train (CV 1-5:)	[0.63, 0.64, 0.63, 0.67, 0.64]
5	V. biologicas y de region	r2 Test (CV 1-5:)	[0.55, 0.49, 0.52, 0.43, 0.51]
6	V. biologicas y de region	neg_mean_absolute_error Train (CV 1-5:)	[-34.21, -33.49, -34.89, -32.84, -35.27]
7	V. biologicas y de region	neg_mean_absolute_error Test (CV 1-5:)	[-39.62, -42.55, -36.82, -44.52, -35.29]
8	V. temporales y de ubicacion	r2 Train (CV 1-5:)	[0.17, 0.19, 0.17, 0.17, 0.17]
9	V. temporales y de ubicacion	r2 Test (CV 1-5:)	[-0.29, -0.33, -0.39, -0.18, -0.34]
10	V. temporales y de ubicacion	neg_mean_absolute_error Train (CV 1-5:)	[-79.85, -77.32, -81.34, -76.97, -82.85]
11	V. temporales y de ubicacion	neg_mean_absolute_error Test (CV 1-5:)	[-98.66, -104.43, -95.36, -103.37, -92.21]

Figura 21. Resultado Random Forest

En base a los resultados se puede observar lo siguiente en base al subconjunto seleccionado y a la métrica R2:

- V. Todas: En el caso de seleccionar todas las variables se observa que los valores de r2 en el entrenamiento se encuentran en el siguiente rango 0.94 y entre 0.51 y 0.65. esto significa que el modelo estaría sobre ajustado.
- V. biológicas y de región: En este caso se observa que los valores de r2 en el entrenamiento se encuentran en el siguiente rango 0.63 y 0.67 y entre 0.43 y 0.55 en test. Esto significa que el modelo no está aprendiendo a predecir.
- V. temporales y de ubicación: En este caso se observa que los valores de r2 en el entrenamiento se encuentran en el siguiente rango 0.17 y 0.19 y con valores negativos en test. Esto sugiere que este conjunto de variables tiene una capacidad nula de predicción.

En Conclusión, respecto a la métrica R2, el conjunto de variables que ofrece los mejores resultados es "V. Todas". Sin embargo, cabe destacar que este modelo estaría sobre ajustado.

5.2.3 MLPRegressor

Se ha aplicado un perceptrón multicapa para poder determinar si es un modelo adecuado. A continuación, se muestran los resultados a la hora de aplicar este algoritmo a nuestro problema.:

	Feature Set	Metric	Score
0	V. Todas	r2 Train (CV 1-5:)	[0.14, 0.27, 0.13, 0.15, 0.12]
1	V. Todas	r2 Test (CV 1-5:)	[0.14, 0.27, 0.15, 0.12, 0.1]
2	V. Todas	neg_mean_absolute_error Train (CV 1-5:)	[-80.85, -58.62, -85.78, -79.41, -91.71]
3	V. Todas	neg_mean_absolute_error Test (CV 1-5:)	[-80.75, -64.13, -79.8, -87.35, -85.4]
4	V. biologicas y de region	r2 Train (CV 1-5:)	[0.41, 0.43, 0.39, 0.46, 0.4]
5	V. biologicas y de region	r2 Test (CV 1-5:)	[0.46, 0.43, 0.48, 0.34, 0.41]
6	V. biologicas y de region	neg_mean_absolute_error Train (CV 1-5:)	[-53.62, -50.28, -56.3, -53.06, -61.08]
7	V. biologicas y de region	neg_mean_absolute_error Test (CV 1-5:)	[-50.66, -56.32, -50.55, -60.53, -56.99]
8	V. temporales y de ubicacion	r2 Train (CV 1-5:)	[0.0, 0.0, 0.0, 0.01, -0.0]
9	V. temporales y de ubicacion	r2 Test (CV 1-5:)	[0.0, -0.0, -0.01, 0.0, -0.01]
10	V. temporales y de ubicacion	neg_mean_absolute_error Train (CV 1-5:)	[-86.13, -86.81, -95.73, -85.25, -91.79]
11	V. temporales y de ubicacion	neg_mean_absolute_error Test (CV 1-5:)	[-85.75, -91.99, -89.61, -94.5, -83.63]

Figura 22. Resultado MLP

En base a los resultados se puede observar lo siguiente en base al subconjunto seleccionado y a la métrica R2:

- V. Todas: En el caso de seleccionar todas las variables se observa que los valores de r2 en el entrenamiento se encuentran en el siguiente rango 0.26 y 0.31 y entre 0.23 y 0.35. Esto significa que el modelo no aprende a predecir.
- V. Biológicas y de región: En este caso se observa que los valores de r2 en el entrenamiento se encuentran en el siguiente rango 0.29 y 0.48 y entre 0.32 y 0.43. en test. Esto significa que el modelo no está aprendiendo a predecir.
- V. temporales y de ubicación: En este caso se observa que los valores de r2 en el entrenamiento y test son 0. Esto sugiere que este conjunto de variables tiene una capacidad nula de predicción.

En Conclusión, respecto a la métrica R2, ninguna combinación ofrece resultados óptimos.

5.2.4 XGBRegressor

Finalmente se ha planteado aplicar un algoritmo basado en boosting. A continuación, podemos observar el resultado:

	Feature Set	Metric	Score
0	V. Todas	r2 Train (CV 1-5:)	[0.93, 0.93, 0.92, 0.93, 0.94]
1	V. Todas	r2 Test (CV 1-5:)	[0.61, 0.56, 0.62, 0.49, 0.45]
2	V. Todas	neg_mean_absolute_error Train (CV 1-5:)	[-19.05, -19.07, -20.68, -19.19, -18.84]
3	V. Todas	neg_mean_absolute_error Test (CV 1-5:)	[-40.51, -46.08, -37.33, -46.53, -40.29]
4	V. biologicas y de region	r2 Train (CV 1-5:)	[0.64, 0.64, 0.63, 0.67, 0.64]
5	V. biologicas y de region	r2 Test (CV 1-5:)	[0.55, 0.46, 0.5, 0.43, 0.51]
6	V. biologicas y de region	neg_mean_absolute_error Train (CV 1-5:)	[-34.54, -33.55, -35.3, -33.1, -35.53]
7	V. biologicas y de region	neg_mean_absolute_error Test (CV 1-5:)	[-40.4, -44.05, -38.37, -44.92, -36.27]
8	V. temporales y de ubicacion	r2 Train (CV 1-5:)	[0.17, 0.19, 0.17, 0.16, 0.16]
9	V. temporales y de ubicacion	r2 Test (CV 1-5:)	[-0.22, -0.37, -0.32, -0.13, -0.28]
10	V. temporales y de ubicacion	neg_mean_absolute_error Train (CV 1-5:)	[-80.77, -78.14, -82.26, -77.84, -83.54]
11	V. temporales y de ubicacion	neg_mean_absolute_error Test (CV 1-5:)	[-95.69, -103.34, -92.53, -101.24, -90.09]

Figura 23. Resultado XGB

En base a los resultados se puede observar lo siguiente en base al subconjunto seleccionado y a la métrica R2:

- V. Todas: En el caso de seleccionar todas las variables se observa que los valores de r2 en el entrenamiento se encuentran en el siguiente rango 0.93 y 0.94 y entre 0.41 y 0.6. Esto significa que el modelo está preajustado.
- V. biológicas y de región: En este caso se observa que los valores de r2 en el entrenamiento se encuentran en el siguiente rango 0.63 y 0.67 y entre 0.43 y 0.55 en test. Esto significa que el modelo está sobre ajustado.
- V. temporales y de ubicación: En este caso se observa que los valores de r2 en el entrenamiento y test son bajos. Esto sugiere que este conjunto de variables tiene una capacidad nula de predicción.

En Conclusión, respecto a la métrica R2, las dos primeras combinaciones ofrecen resultados buenos en el entrenamiento pero bajos en test por lo que estaría el modelo sobre ajustado.

5.2.5 NN - Modelo secuencial

Por último, se ha querido realizar un análisis añadiendo una red neuronal. Para ello usando la librería de keras se ha creado un modelo secuencial sencillo tal y como se puede observar.

```
model = Sequential()
model.add(Dense(128, input_dim=x_train.shape[1], activation='relu'))
model.add(Dense(64, activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(1))

model.compile(loss='mean_squared_error',
              optimizer='adam',
              metrics=[MeanAbsoluteError(), MeanSquaredError()])
model.fit(x_train, y_train, epochs=50, batch_size=32, verbose=0)
```

Figura 24. Creación de la NN

el resultado obtenido sin usar CV es el siguiente: (Test R2: 0.28) el cual se puede decir que no es adecuado para nuestro objetivo

5.2.6 Comparativa

Tal y como se puede observar en la tabla inferior se puede decir que ningún modelo llega a realizar el objetivo de forma óptima. Se puede decir que los dos mejores modelos serían Random Foorest o XGBRegressor de los cuales destacar que podrían estar sobreajustados.

Model	R2 Train	R2 Test
DecisionTreeRegressor	0.54	0.52
Random Forest	0.94	0.65
MLPRegressor	0.14	0.12
XGBRegressor	0.93	0.62

Tabla 2. Comparativa de los algoritmos de regresión con sus valores R2

5.3 Tarea de clasificación (Label-encoding)

En este caso se ha seleccionado la variable “MPA_Status” el cual representa el estado de la zona marina protegida, es decir el estado de conservación de la zona en la que se recogieron los datos. Este análisis predictivo tiene como objetivo en primera instancia el determinar de qué características depende el estado marcado de la zona protegida lo que

permitiría en un futuro en base a esas características poder predecir el estado de la zona protegida.

Para ello en primer lugar se realizará un análisis de detección de patrones para determinar la correlación entre la variable objetivo y diferentes posibles combinaciones de características. Con esto se conseguirá determinar qué características maximizaran la predicción. En primer lugar, podríamos estudiar las siguientes características:

- year y month: El año podrían tener un impacto en 'MPA_Status' debido a las variaciones en el tiempo de las políticas de conservación, las condiciones climáticas, etc.
- Sum_ind: Esta característica representa la cantidad de organismos individuales y podría tener relación con la variable a predecir.
- Latitude y Longitude: La ubicación podría ser un factor importante en cuanto a la relevancia geográfica en determinado estado de conservación.
- depth_strata: La profundidad podría ser condicionante.
- Island_encoded, Bioregion_encoded, Site_encoded: cada una de las islas/biorregiones o el sitio pueden tener diferentes condiciones y políticas que afecten a la variable objetivo.
- PhylumOrDivision_encoded, Class_encoded, Order_encoded, Family_encoded, ScientificName_encoded: los datos relacionados con la taxonomía de las especies podrían ser determinantes ya que representan la biodiversidad, un factor relevante en el estado de conservación.
- epoca_encoded: La época del año podría afectar el estado de conservación.

Ahora que se tiene una visión global de las diferentes características se podría realizar un análisis más minucioso. Para ello comenzaremos con un análisis que detecte cualquier correlación lineal además de un análisis de agrupamiento mediante cluster para poder determinar otro tipo de relaciones no lineales. Una vez hecho esto ya se podrían seleccionar determinados grupos de características para el análisis predictivo.

5.3.1 Detección de patrones

En primer lugar se muestra la correlación lineal existente entre las demás variables y la variable objetivo mediante la matriz de correlación la cual se puede observar a continuación:

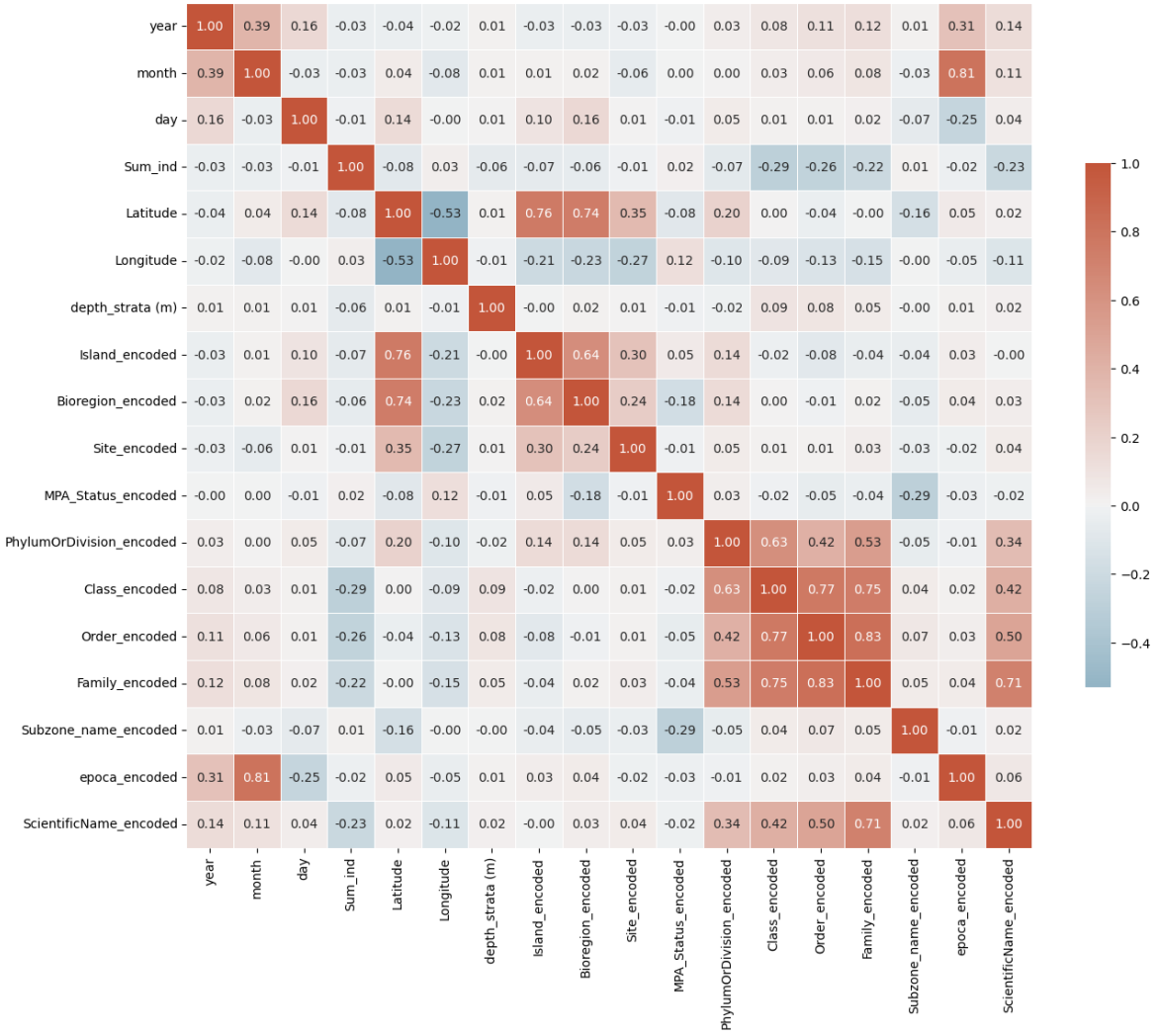

Figura 25. Matriz de confusión

Partiendo de los resultados obtenidos, podemos observar que la columna que estamos tratando de predecir, MPA_Status_encoded, no tiene una correlación fuerte con las demás variables, lo que supondría que podría ser más difícil predecir MPA_Status_encoded utilizando sólo estas variables, además de que la relación entre MPA_Status_encoded y las demás variables no sean lineales, por lo que una matriz de correlación no lo podría capturar. Pese a lo anterior, algunas correlaciones que podemos destacar son las siguientes:

- Subzone_name_encoded: Esta variable tiene una correlación negativa moderada (-0.287574) con MPA_Status_encoded. Esta correlación sugiere que la subzona puede tener cierto efecto en el estado de MPA, aunque esta relación es inversa.
- Island_encoded: Esta variable muestra una correlación débil (0.052732) con MPA_Status_encoded, aunque podría aportar información útil. Bioregion_encoded una correlación negativa.
- Bioregion_encoded: Al igual que Island_encoded, esta variable muestra una correlación moderada débil (-0.175128), aun así, podría aportar información útil al análisis.
- Longitude: Esta variable tiene una correlación moderada positiva (0.121003) con MPA_Status_encoded. Aunque la correlación no sea muy fuerte, podría aportar alguna relación útil aquí para la predicción.

Como conclusión cabe destacar que la correlación puede indicar una relación potencial entre dos variables, pero no implica causalidad y la matriz de correlación sólo nos permite analizar relaciones lineales. Para relaciones no lineales o interacciones entre variables, podríamos requerir análisis adicionales y/o modelos más complejos. A continuación, se explorará la técnica de agrupamiento mediante clusters para poder determinar otro tipo de relación seleccionando previamente un subconjunto de variables.

En primer lugar, mostraremos los resultados de este análisis en nuestro conjunto de datos:

- Variables: ['Latitude', 'Longitude', 'depth_strata (m)']:

MPA_Status_encoded	0	1
Cluster		
0	396	376
1	900	800
2	566	799
3	888	792
4	541	822

Figura 26. Matriz de contingencia para las variables especificadas

- Variables: ['year', 'month', 'epoca_encoded']:

MPA_Status_encoded	0	1
Cluster		
0	416	467
1	1611	1739
2	356	268
3	450	596
4	458	519

Figura 27. Matriz de contingencia para las variables especificadas

- Variables: ['PhylumOrDivision_encoded', 'Class_encoded', 'Order_encoded', 'Family_encoded']:

MPA_Status_encoded	0	1
Cluster		
0	1441	1695
1	1156	1079
2	226	298
3	275	213
4	193	304

Figura 28. Matriz de contingencia para las variables especificadas

- Variables: ['Island_encoded', 'Bioregion_encoded', 'Site_encoded']:

MPA_Status_encoded	0	1
Cluster		
0	970	1028
1	756	882
2	788	771
3	474	757
4	303	151

Figura 29. Matriz de contingencia para las variables especificadas

Según los resultados que hemos obtenido, podemos observar que la elección de variables influye en cómo se forman los clusters además de cómo se distribuye el 'MPA_Status_Encoded' dentro de cada cluster. Para nuestro análisis hemos hecho las siguientes combinaciones:

- Combinación 1: ['Latitude', 'Longitude', 'depth_strata (m)']: Esta agrupación se basa en los datos geográficos de longitud, latitud y profundidad. Se puede observar que la distribución de ' MPA_Status_Encoded ' está bastante equilibrada

en la mayoría de los clusters, sin embargo, podemos observar que el cluster 1 tiene más áreas marinas protegidas con estado "0" y los clusters 2 y 3 tienen más áreas marinas protegidas con estado "1", esto podría sugerir que ciertas áreas geográficas o ciertos niveles de profundidad pueden estar más o menos relacionados con diferentes estados de áreas marinas protegidas.

- Combinación 2: ['year', 'month', 'epoca_encoded']: Esta segunda agrupación se basa en los datos temporales. En este caso, se puede observar que el cluster 1 tiene un número significativamente mayor de observaciones, esto podría indicar que hay ciertos periodos de tiempo que tienen un alto número de observaciones y, por otra parte, el cluster 2 tiene una proporción mayor de "0", lo que podría sugerir que, en ciertos periodos de tiempo, es más probable que las áreas marinas protegidas tengan un estado "0".
- Combinación 3: ['PhylumOrDivision_encoded', 'Class_encoded', 'Order_encoded', 'Family_encoded']: Esta agrupación se basa en las características biológicas de las áreas marinas protegidas, podemos observar que el cluster 1 vuelve a tener un número significativamente mayor de observaciones, lo que podría suponer que hay ciertos grupos taxonómicos que son más comunes en las áreas marinas protegidas, por otra parte, los clusters 2, 3 y 4 tienen una distribución equilibrada de 'MPA_Status', mientras que el cluster 0 tiende a tener más 'MPA_Status' con estado "1". Esto podría sugerir que ciertos grupos taxonómicos pueden estar más asociados con un estado particular de las áreas marinas protegidas.
- Combinación 4: ['Island_encoded', 'Bioregion_encoded', 'Site_encoded']: Esta última agrupación se basa en la ubicación específica de las áreas marinas protegidas. La distribución de 'MPA_Status' está equilibrada en los clusters 0, 1 y 2, sin embargo, el cluster 3 tiene una proporción ligeramente mayor de "1", y por otro lado el cluster 4 tiene una proporción bastante mayor de "0". Esto podría sugerir que ciertas ubicaciones específicas pueden estar más asociadas con un estado particular de áreas marinas protegidas.

Ahora bien, para poder realizar el análisis predictivo y en base al análisis anterior se puede deducir que con las variables biológicas y de sitio se observa una separación significativa en los clusters, lo que sugiere que estas variables pueden ser útiles para la predicción. Cabe destacar que los clusters se diferenciaban en este caso principalmente en el número

de observaciones y no en la proporción del MPA_Status. Esta variabilidad en los clusters podría indicar que diferentes grupos de especies podrían estar relacionadas con diferentes estados de las MPA. Una posible respuesta podría relacionarse con factores ecológicos, como la diversidad de especies, características del hábitat que podrían influir en las decisiones sobre la protección de las áreas marinas.

En cuanto a las variables relacionadas con la ubicación y el tiempo, se puede observar separación entre los cluster pero no muy marcada como en el caso anterior. Esto no quiere decir que no estas variables no influyen en el resultado.

Dado ambos análisis realizados a la hora de seleccionar un modelo de predicción y entrenarlo, se podrían considerar todas las variables mencionadas. Dicho esto, se probará con diferentes combinaciones y así poder concluir que diferentes características influyen en la variable objetivo.

5.1.2 KNN

	Feature Set	Metric	Score
0	V. biologicas y de region	accuracy Train (CV 1-5:)	[0.98, 0.98, 0.98, 0.98, 0.98]
1	V. biologicas y de region	accuracy Test (CV 1-5:)	[0.93, 0.96, 0.93, 0.97, 0.92]
2	V. biologicas y de region	f1_macro Train (CV 1-5:)	[0.98, 0.98, 0.98, 0.98, 0.98]
3	V. biologicas y de region	f1_macro Test (CV 1-5:)	[0.93, 0.96, 0.92, 0.97, 0.92]
4	V. temporales y de ubicacion	accuracy Train (CV 1-5:)	[1.0, 1.0, 1.0, 1.0, 1.0]
5	V. temporales y de ubicacion	accuracy Test (CV 1-5:)	[0.61, 0.72, 0.6, 0.75, 0.66]
6	V. temporales y de ubicacion	f1_macro Train (CV 1-5:)	[1.0, 1.0, 1.0, 1.0, 1.0]
7	V. temporales y de ubicacion	f1_macro Test (CV 1-5:)	[0.61, 0.71, 0.59, 0.75, 0.66]
8	V. Todas	accuracy Train (CV 1-5:)	[0.91, 0.9, 0.9, 0.91, 0.91]
9	V. Todas	accuracy Test (CV 1-5:)	[0.79, 0.78, 0.78, 0.77, 0.71]
10	V. Todas	f1_macro Train (CV 1-5:)	[0.91, 0.9, 0.9, 0.91, 0.91]
11	V. Todas	f1_macro Test (CV 1-5:)	[0.79, 0.78, 0.78, 0.77, 0.7]

Figura 30. Resultado KNN

Partiendo de los resultados obtenidos del algoritmo KNN, podemos observar que las diferentes combinaciones de variables dan resultados variados, según la métrica de precisión.

- Variables Biológicas y de Región: Los valores de accuracy obtenidos sobre el conjunto de entrenamiento son bastante altos, con un promedio de 0.98. Por otra parte, en el conjunto de test, los valores son generalmente buenos, aunque muestran una variabilidad más grande que va desde 0.92 hasta 0.97.

- Variables Temporales y de Ubicación: Esta combinación de variables da lugar a un modelo con resultados perfectos sobre el conjunto de entrenamiento, con una precisión de 1.0, sin embargo, el accuracy en los en el conjunto de test, es bastante más baja, con resultados entre 0.61 y 0.75, por lo que podemos suponer que es un indicio de sobreajuste, ya que el modelo se desempeña perfectamente en el conjunto de entrenamiento, pero no llega a generalizar a nuevos datos.
- Todas las Variables: En este caso, el accuracy es menor tanto para el entrenamiento como para la prueba en comparación con las otras dos combinaciones de variables. Los valores de accuracy para el conjunto de entrenamiento varían de 0.90 a 0.91, mientras que para el conjunto de test da resultados entre 0.71 y 0.79. Aunque los valores de accuracy son más bajos, la brecha entre los resultados de los conjuntos de entrenamiento y test es más pequeña que la combinación anterior, lo que sugiere que este modelo puede estar menos sobreajustado.

5.1.3 Decision Tree

	Feature Set	Metric	Score
0	V. biologicas y de region	accuracy Train (CV 1-5:)	[1.0, 1.0, 1.0, 1.0, 1.0]
1	V. biologicas y de region	accuracy Test (CV 1-5:)	[0.97, 0.99, 0.97, 1.0, 1.0]
2	V. biologicas y de region	f1_macro Train (CV 1-5:)	[1.0, 1.0, 1.0, 1.0, 1.0]
3	V. biologicas y de region	f1_macro Test (CV 1-5:)	[0.97, 0.99, 0.97, 1.0, 1.0]
4	V. temporales y de ubicacion	accuracy Train (CV 1-5:)	[1.0, 1.0, 1.0, 1.0, 1.0]
5	V. temporales y de ubicacion	accuracy Test (CV 1-5:)	[0.98, 0.99, 0.97, 1.0, 1.0]
6	V. temporales y de ubicacion	f1_macro Train (CV 1-5:)	[1.0, 1.0, 1.0, 1.0, 1.0]
7	V. temporales y de ubicacion	f1_macro Test (CV 1-5:)	[0.98, 0.99, 0.97, 1.0, 1.0]
8	V. Todas	accuracy Train (CV 1-5:)	[1.0, 1.0, 1.0, 1.0, 1.0]
9	V. Todas	accuracy Test (CV 1-5:)	[0.98, 0.99, 0.97, 1.0, 0.98]
10	V. Todas	f1_macro Train (CV 1-5:)	[1.0, 1.0, 1.0, 1.0, 1.0]
11	V. Todas	f1_macro Test (CV 1-5:)	[0.98, 0.99, 0.97, 1.0, 0.98]

Figura 31. Resultado Decision Tree

Partiendo de los resultados obtenidos del algoritmo decision tree, podemos observar que las diferentes combinaciones de variables resultados más uniformes que el algoritmo anterior de KNN, según la métrica de precisión.

- Variables Biológicas y de Región: Los valores de accuracy obtenidos sobre el conjunto de entrenamiento perfectos, con un promedio de 1.00. En el conjunto de test, los valores son bastante buenos también, con un promedio de 0.98. Como

podemos observar, este modelo se ajusta bastante bien a los conjuntos y generaliza bastante bien también.

- Variables Temporales y de Ubicación: Con esta combinación obtenemos un modelo que también se ajusta a la perfección a los datos del conjunto de entrenamiento, con un promedio de 1.0. y al igual que el conjunto anterior, obtiene unos resultados rozando la perfección en el conjunto de test, con un promedio de 0.99. Al igual que la combinación anterior también podemos concluir que el modelo generaliza bastante bien.
- Todas las Variables: Con esta combinación obtenemos un modelo que también se ajusta a la perfección a los datos del conjunto de entrenamiento, con un promedio de 1.0. y al igual que el conjunto anterior, obtiene unos resultados rozando bastante buenos en el conjunto de test, con un promedio de 0.98. Al igual que las 2 combinaciones anteriores, también podemos concluir que el modelo generaliza bastante bien.

5.1.4 Bagging

	Feature Set	Metric	Score
0	V. biologicas y de region	accuracy Train (CV 1-5:)	[0.81, 0.83, 0.78, 0.82, 0.8]
1	V. biologicas y de region	accuracy Test (CV 1-5:)	[0.77, 0.83, 0.7, 0.81, 0.83]
2	V. biologicas y de region	f1_macro Train (CV 1-5:)	[0.8, 0.82, 0.77, 0.82, 0.8]
3	V. biologicas y de region	f1_macro Test (CV 1-5:)	[0.76, 0.83, 0.69, 0.81, 0.83]
4	V. temporales y de ubicacion	accuracy Train (CV 1-5:)	[0.98, 0.99, 0.91, 0.98, 0.89]
5	V. temporales y de ubicacion	accuracy Test (CV 1-5:)	[0.91, 0.99, 0.87, 0.99, 0.87]
6	V. temporales y de ubicacion	f1_macro Train (CV 1-5:)	[0.98, 0.99, 0.91, 0.98, 0.89]
7	V. temporales y de ubicacion	f1_macro Test (CV 1-5:)	[0.91, 0.99, 0.87, 0.99, 0.87]
8	V. Todas	accuracy Train (CV 1-5:)	[0.96, 0.95, 0.96, 0.97, 0.95]
9	V. Todas	accuracy Test (CV 1-5:)	[0.93, 0.95, 0.92, 0.98, 0.92]
10	V. Todas	f1_macro Train (CV 1-5:)	[0.96, 0.95, 0.96, 0.97, 0.95]
11	V. Todas	f1_macro Test (CV 1-5:)	[0.93, 0.95, 0.92, 0.98, 0.92]

Figura 32. Resultado Bagging

Partiendo de los resultados obtenidos usando bagging con estimador de decision tree.

- Variables Biológicas y de Región: Los valores de accuracy obtenidos sobre el conjunto de entrenamiento son notablemente menores que los resultados obtenidos con KNN y decisión tree, con un promedio de 0.80. En el conjunto de test, los valores obtenidos son parecidos, con un promedio de 0.81. Podemos observar que este modelo no llega a ajustarse del todo bien al conjunto.

- Variables Temporales y de Ubicación: Con esta combinación obtenemos un modelo que también se ajusta bastante bien a los datos del conjunto de entrenamiento, con un promedio de 0.95. En el conjunto de test, se obtienen unos resultados rozando bastante buenos también, con un promedio de 0.92. Con esta combinación podemos observar que el modelo generaliza bastante bien.
- Todas las Variables: Con esta combinación obtenemos un modelo que también se ajusta muy bien a los datos del conjunto de entrenamiento, con un promedio de 0.97. Y al igual que la combinación anterior, obtiene unos resultados bastante buenos en el conjunto de test, con un promedio de 0.95. Al igual que la combinación anterior, también podemos concluir que el modelo generaliza bastante bien.

5.1.5 Boosting

	Feature Set	Metric	Score
0	V. biologicas y de region	accuracy Train (CV 1-5:)	[1.0, 1.0, 1.0, 1.0, 1.0]
1	V. biologicas y de region	accuracy Test (CV 1-5:)	[0.97, 0.99, 0.97, 1.0, 0.98]
2	V. biologicas y de region	f1_macro Train (CV 1-5:)	[1.0, 1.0, 1.0, 1.0, 1.0]
3	V. biologicas y de region	f1_macro Test (CV 1-5:)	[0.97, 0.99, 0.97, 1.0, 0.98]
4	V. temporales y de ubicacion	accuracy Train (CV 1-5:)	[1.0, 1.0, 1.0, 1.0, 1.0]
5	V. temporales y de ubicacion	accuracy Test (CV 1-5:)	[0.98, 0.99, 0.97, 1.0, 1.0]
6	V. temporales y de ubicacion	f1_macro Train (CV 1-5:)	[1.0, 1.0, 1.0, 1.0, 1.0]
7	V. temporales y de ubicacion	f1_macro Test (CV 1-5:)	[0.98, 0.99, 0.97, 1.0, 1.0]
8	V. Todas	accuracy Train (CV 1-5:)	[1.0, 1.0, 1.0, 1.0, 1.0]
9	V. Todas	accuracy Test (CV 1-5:)	[0.98, 0.99, 0.97, 1.0, 1.0]
10	V. Todas	f1_macro Train (CV 1-5:)	[1.0, 1.0, 1.0, 1.0, 1.0]
11	V. Todas	f1_macro Test (CV 1-5:)	[0.98, 0.99, 0.97, 1.0, 1.0]

Figura 33. Resultado Boosting

Partiendo de los resultados obtenidos usando Adabost y decision tree como estimador.

- Variables Biológicas y de Región: Los valores de accuracy obtenidos sobre el conjunto de entrenamiento son perfectos, con un promedio de 1.00. En el conjunto de test, los valores son bastante buenos también, con un promedio de 0.98. Como podemos observar, este modelo se ajusta bastante bien a los conjuntos y generaliza bastante bien también.
- Variables Temporales y de Ubicación: Con esta combinación obtenemos un modelo que también se ajusta a la perfección a los datos del conjunto de entrenamiento, con un promedio de 1.0. y al igual que el conjunto anterior, obtiene unos resultados rozando la perfección en el conjunto de test, con un promedio de

0.99. Al igual que la combinación anterior también podemos concluir que el modelo generaliza bastante bien.

- Todas las Variables: Con esta combinación obtenemos un modelo que también se ajusta a la perfección a los datos del conjunto de entrenamiento, con un promedio de 1.0. y al igual que el conjunto anterior, obtiene unos resultados rozando bastante buenos en el conjunto de test, con un promedio de 0.98. Al igual que las 2 combinaciones anteriores, también podemos concluir que el modelo generaliza bastante bien.

5.1.6 Stacking

	Feature Set	Metric	Score
0	V. biologicas y de region	accuracy Train (CV 1-5:)	[0.98, 0.98, 0.98, 0.98, 0.98]
1	V. biologicas y de region	accuracy Test (CV 1-5:)	[0.94, 0.97, 0.93, 0.97, 0.95]
2	V. biologicas y de region	f1_macro Train (CV 1-5:)	[0.98, 0.98, 0.98, 0.98, 0.98]
3	V. biologicas y de region	f1_macro Test (CV 1-5:)	[0.94, 0.97, 0.93, 0.97, 0.95]
4	V. temporales y de ubicacion	accuracy Train (CV 1-5:)	[0.99, 0.81, 0.77, 0.78, 0.78]
5	V. temporales y de ubicacion	accuracy Test (CV 1-5:)	[0.77, 0.77, 0.76, 0.75, 0.79]
6	V. temporales y de ubicacion	f1_macro Train (CV 1-5:)	[0.99, 0.8, 0.77, 0.77, 0.77]
7	V. temporales y de ubicacion	f1_macro Test (CV 1-5:)	[0.77, 0.76, 0.75, 0.73, 0.78]
8	V. Todas	accuracy Train (CV 1-5:)	[0.95, 0.94, 0.89, 0.95, 0.94]
9	V. Todas	accuracy Test (CV 1-5:)	[0.89, 0.91, 0.84, 0.93, 0.94]
10	V. Todas	f1_macro Train (CV 1-5:)	[0.95, 0.94, 0.89, 0.95, 0.94]
11	V. Todas	f1_macro Test (CV 1-5:)	[0.89, 0.91, 0.84, 0.92, 0.94]

Figura 34. Resultado Stacking

Partiendo de los resultados obtenidos usando stacking y decision tree y KNN como estimadores.

- Variables Biológicas y de Región: Los valores de accuracy obtenidos sobre el conjunto de entrenamiento son bastante buenos, con un promedio de 0.98. En el conjunto de test, los valores son bastante buenos también, con un promedio de 0.95. Como podemos observar, este modelo se ajusta bastante bien a los conjuntos y generaliza bastante bien también.
- Variables Temporales y de Ubicación: Con esta combinación obtenemos un modelo que da unos resultados muy variados, entre 0.77 y 0.99. En el conjunto de test, se obtienen resultados más uniformes pero algo bajos, sobre el 0.77. Según estos resultados podemos observar que este modelo no llega a adaptarse bien sobre el conjunto.

- Todas las Variables: Con esta combinación obtenemos un modelo que también se ajusta bien a los datos del conjunto de entrenamiento, con un promedio de 0.93. Y al igual que en el conjunto de entrenamiento, en el conjunto de test, se obtienen unos resultados bastante buenos, con un promedio de 0.90.

5.1.7 Comparativa

A continuación, se mostrará el mejor resultado obtenido para cada uno de los modelos sin importar la combinación de variables seleccionadas:

Model	Accuracy Train	Accuracy Test
KNN	0.98	0.97
DT	1	0.98
Bagging	0.99	0.98
Boosting	1	0.98
Stacking	0.98	0.98

Tabla 1¡3. Comparativa de los algoritmos de clasificación con sus valores accuracy

Se puede deducir que para este problema se ha podido desarrollar un modelo óptimo que cumple con el objetivo ya que todos ellos presentan muy buenos resultados.

7. CONCLUSIONES

A partir del desarrollo del presente proyecto de minería de datos, se realizó un análisis de patrones submarinos en especies de macroinvertebrados de monitoreo ecológico utilizando el dataset proporcionado por la Fundación Charles Darwin. A través del procesamiento, análisis de los datos e implementación de diferentes algoritmos, se llegaron a diversas conclusiones.

En primer lugar, cabe destacar que el dataset estudiado recoge información sobre diferentes variables, incluyendo las relacionadas a ubicación geográfica, características

taxonómicas y abundancia de especies. Dada la naturaleza de este conjunto de datos, se han implementado diversas técnicas para poder determinar diferentes patrones entre las diferentes características. Además, se han aplicado diferentes algoritmos y diferentes ajustes en estos para poder realizar el análisis predictivo con el fin de obtener el modelo más óptimo. Sin embargo, en el caso de la tarea de regresión, los resultados no han sido tan satisfactorios como se esperaba. Por otro lado, en cuanto a la clasificación, se ha observado que distintos modelos han obtenido resultados óptimos.

En general, este trabajo ha logrado cumplir con los objetivos establecidos para el proyecto, ya que se han identificado patrones y características de las especies de macroinvertebrados marinos en relación con diversas variables ambientales y taxonómicas. Sin embargo, con el objetivo de mejorar los resultados de este análisis, sería interesante considerar nuevas soluciones que no se han abordado en este informe. Estas soluciones se detallan en la siguiente sección.

8. FUTURAS INVESTIGACIONES

Durante el desarrollo de esta práctica, se logró identificar información relevante y explorar diversas alternativas para abordar el objeto de estudio. No obstante, es importante señalar que aún existen muchas preguntas sin respuesta y áreas de investigación por explorar. En este sentido, se presentan algunas sugerencias para futuras investigaciones que podrían contribuir a la mejora de los resultados obtenidos en este análisis:

- Exploración de nuevas variables: Contar con características adicionales podría proporcionar información adicional y relevante para ayudar al modelo a capturar mejor las relaciones y patrones subyacentes en los datos. Para este contexto, las variables podrían estar relacionadas con factores ambientales, características genéticas de las especies, entre otras.

- Mejorar la calidad de los datos: Con el objetivo de intentar obtener mejores resultados, en futuras investigaciones podría evaluarse la necesidad de realizar una revisión aún más exhaustiva del dataset, consultando con otros expertos en el

dominio de ser necesario. La calidad de los datos es fundamental para obtener resultados precisos y confiables de un modelo de aprendizaje automático, por lo tanto, invertir tiempo y esfuerzo en mejorar la calidad de la información en el dataset puede marcar la diferencia en el rendimiento del modelo.

- Estudios comparativos: Realizar estudios comparativos con otros grupos de investigación que trabajen con información similar podría ser de utilidad, ya que se podrían identificar patrones compartidos o diferencias relevantes ayuden a las futuras investigaciones.