Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

Predicción de tipo de buque utilizando información de áreas de actividad y técnicas de inteligencia artificial

Grado en Ingeniería Mecánica

ALUMNO: Ignacio de Gandarillas Carrara

DIRECTORES: Belén Barragáns Martínez

Pablo Sendín Raña

CURSO ACADÉMICO: 2022-2023

Universidade Vigo

Centro Universitario de la Defensa
en la Escuela Naval Militar

TRABAJO FIN DE GRADO

Predicción de tipo de buque utilizando información de áreas de actividad y técnicas de inteligencia artificial

Grado en Ingeniería Mecánica

Intensificación en Tecnología Naval

Cuerpo General

Universidade Vigo

RESUMEN

Este Trabajo Fin de Grado se enmarca dentro de una línea de investigación del Centro Universitario de la Defensa en la Escuela Naval Militar (CUD-ENM) relacionada con la aplicación de técnicas de inteligencia artificial para mejorar el conocimiento del entorno marítimo. En un TFG anterior se exploró la capacidad de predecir el campo de los mensajes AIS que identifica el tipo de buque (campo especialmente relevante y que, en muchos casos, no viene cubierto) empleando otros datos de los mensajes, en particular, aquellos estáticos, relacionados con las dimensiones del buque, o los dinámicos, relacionados con su velocidad o rumbo.

En este TFG se plantea el análisis del impacto en la calidad de la predicción del tipo de buque que supondría el empleo de información relacionada con el área de actividad del barco. Para ello, se estudiará cómo incorporar dicha información como entrada a los algoritmos de aprendizaje supervisado empleados y se presentarán todas las combinaciones de experimentos llevados a cabo.

El TFG concluye que la información del área de actividad del buque contribuye positivamente a mejorar la predicción obtenida, señalándose aquellos tipos de barcos que se benefician en mayor medida del empleo de esta información por el algoritmo de clasificación.

PALABRAS CLAVE

Inteligencia artificial, Tipo de buque, *Random Forest,* Celdas H3, Datos AIS

AGRADECIMIENTOS

En primer lugar, me gustaría agradecer a mi promoción por acompañarme durante todos estos años. Han sido cinco largos años que se han pasado volando, llenos de momentos buenos y no tan buenos, pero de los que siempre hemos aprendido algo. Llegamos aquí siendo adolescentes completamente desconocidos y nos vamos siendo adolescentes más mayores, pero siendo una gran familia que espero y deseo me acompañe el resto de mi vida. Gracias.

A mis tutores, Belén y Pablo. Agradecerles la dedicación plena que han tenido durante estos meses. Desde el primer día han estado completamente disponibles para ayudarme y guiarme en el desarrollo de este trabajo. Su ilusión y amor por su trabajo me ha hecho adquirir no solamente conceptos técnicos, sino también, otros aspectos que incluso considero son más importantes; con paciencia, entusiasmo, esfuerzo, dedicación y amor por el trabajo se pueden conseguir grandes cosas. Ellos son un gran ejemplo para esta Escuela. Gracias.

Y, por último, quiero agradecérselo a toda mi familia, porque siempre están disponibles para lo que necesito, son el pilar fundamental de mi vida, gracias por inculcarme los valores que tengo y por enseñarme lo verdaderamente importante de la vida. En particular, quiero agradecérselo a mis padres, Jaime y María, sois mi ejemplo y mi guía en el camino. Hoy, soy lo que soy gracias a vosotros.

Gracias.

CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1 INTRODUCCIÓN Y OBJETIVOS

1.1 Contexto y motivación

Es inimaginable pensar en tecnología sin hacer referencia a la inteligencia artificial. En los últimos años esta rama de la ciencia ha tomado una gran importancia en muchos ámbitos diferentes, desde robots de cocina, coches autónomos hasta en la medicina. La Armada como empresa innovadora tampoco ha querido quedarse atrás. Es uno de sus objetivos principales el desarrollo y adaptación de la Flota a los nuevos tiempos, incorporando equipos y medios de última generación con el fin de tener unidades eficientes.

En las líneas generales de la Armada para 2022 [1] el Almirante General Jefe de Estado Mayor de la Armada (AJEMA) marca como propósito general que la Armada siga siendo decisiva y relevante. Para ello expone diferentes principios, entre los que se destaca la relevancia de la Transformación Digital como medio para obtener la agilidad deseada de la organización. Además, también subraya la importancia de seguir explorando nuevos ámbitos para mantener la superioridad tecnológica frente a la de los adversarios, e invertir en investigación, desarrollo e innovación.

Motivado por estas directrices nacen varios proyectos entre el que destaca el proyecto [2] donde se encuadra este TFG que tienen el fin de mejorar el Conocimiento del Entorno Marítimo (CEM) mediante detección de anomalías, haciendo uso de técnicas de inteligencia artificial.

Dicho proyecto fue solicitado por la Armada al Centro Universitario de la Defensa en la Escuela Naval Militar (CUD-ENM). La institución encargada del CEM en la Armada es el Centro de Operaciones y Vigilancia de Acción Marítima (COVAM). Entre sus funciones están mantener una *Recognise Maritime Picture* (RMP) robusta. Para todo ello en el CUD-ENM se desarrollaron un proyecto; CEMAI (Inteligencia Artificial para el Conocimiento del Entorno Marítimo) con el que se desarrolla un demostrador que permitía la aplicación de Inteligencia Artificial (IA) con el propósito de mejorar los procedimientos operativos utilizados por el COVAM.

Los datos que se utilizaron fueron obtenidos de los sistemas AIS (*Automatic Identification System*). Que es la fuente de información que permite conocer la posición de los barcos en todo el globo terráqueo. No solamente como sistema de seguridad para los propios barcos (la Organización Marítima Internacional, OMI, obliga su utilización a la mayor parte de ello), sino que también sirve como sistema para mantener las diferentes zonas marítimas controladas por las respectivas autoridades. La información que ofrecen dichos mensajes se puede subdividir en dos campos principales: información relativa a las características del barco e información cinemática. El primer campo está formado por todas las características propias de cada barco como son el nombre del barco, la nacionalidad, bandera, la eslora, la manga, el calado… Por otro lado, se encuentran los parámetros cinemáticos que dan información relativa a la situación del barco en cada momento: velocidad, rumbo y localización geográfica

Los resultados mostraron que el demostrador era capaz de procesar el flujo AIS en tiempo real y detectar comportamientos anómalos. Pero se reseñó que una gran parte de las veces los datos AIS que se recibían estaban incompletos o eran falsos (en particular, un tercio de los datos no incluía el valor del tipo de barco). La Figura 1-1 muestra una situación normal del Mar Mediterráneo. Como se puede observar en la imagen el número de barcos que se encuentran navegando es muy elevado, sí aproximadamente una tercera parte no lleva cubierto el campo de tipo de barco, el número de barcos sospechosos sería muy elevado, lo que se traduciría en un gran número de barcos que estudiar que materialmente es imposible si no se conoce información de ellos.

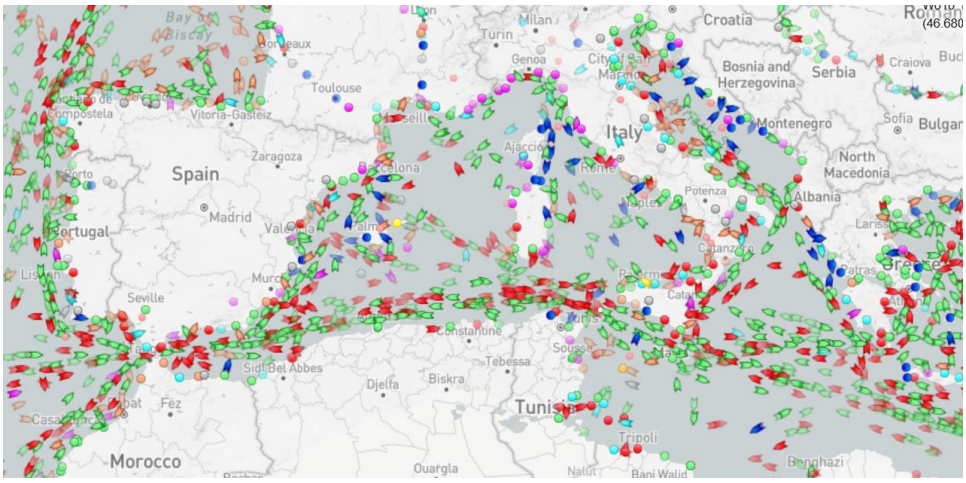


Figura 1-1. Situación marítima del Mar Mediterráneo (fuente: [3])

También se definió en dicho proyecto que este parámetro es uno de los más relevantes a la hora de identificar los datos anómalos. El tipo de barco ayuda a diferenciar el comportamiento que tiene que llevar un buque en base a su clase, en el siguiente ejemplo se muestra la influencia de este parámetro: un mercante navega con una velocidad constante de 20 nudos, en un momento determinado reduce su velocidad de 20 nudos a 3 nudos durante un par de horas y pasado este tiempo retoma su velocidad inicial. Conocida la clase de buque se puede entender que la reducción de velocidad durante ese periodo de tiempo no tiene lógica, en cambio si se estuviese analizando un barco cuya clase fuese un pesquero podría ser más coherente debido a sus funciones. Es por ello por lo que el campo de tipo de buque es tan importante para la identificación de anomalías.

En relación con este proyecto se realizó durante el curso 2021-2022 un Trabajo de Fin de Grado que tenía como fin la predicción de tipo de buque en base a técnicas de inteligencia artificial [4]. En él se desarrolló un modelo de predicción basado fundamentalmente en los parámetros estáticos del buque, llegando a la conclusión de que los parámetros cinemáticos eran menos eficientes en la predicción. Es por ello por lo que en este trabajo se va a enfocar en mejorar la predicción de tipo de buque utilizando información de áreas de actividad y técnicas de inteligencia artificial. Para ello se utilizará información de los datos AIS, así como información adicional generada en el CUD-ENM, en particular, las celdas H3. Mediante la plataforma Big Data para el tratamiento de datos AIS del CUD-ENM se analizan las muestras, se depuran y se enriquecen con otras fuentes de datos (bases de datos registrales, zonas de fondeo, celdas H3, etc).

Por otro lado, se quiere incluir un nuevo parámetro de análisis para la predicción de tipo de buque, las áreas de actividad representadas con las celdas H3. Se trata de son un sistema de indexación geoespacial utilizadas por Uber [5] que divide el espacio en hexágonos de diferentes tamaños en base al nivel de resolución. Mediante la utilización de celdas de diferentes niveles se puede llegar a definir patrones de los diferentes tipos de barco. Por ejemplo, como se puede ver en la Figura 1-2, dos barcos de diferente clase, ambos navegando en la misma celda H3 (roja) cuyos patrones podrían ser iguales, si se toma una celda de resolución mayor (verde) se podría distinguir que el mercante va cambiando de celdas mientras que el pesquero está en una zona de pesca (celdas azules).

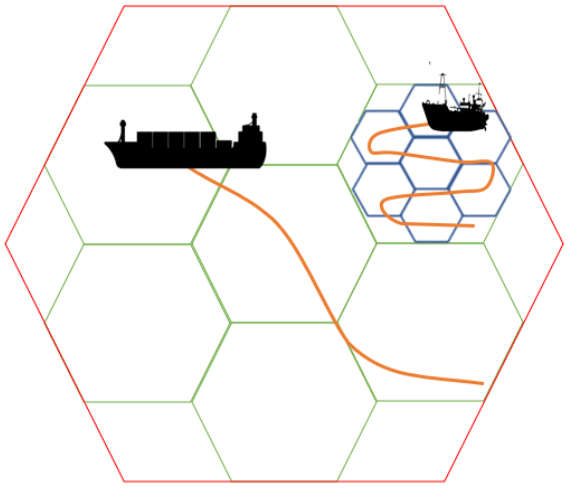


Figura 1-2. Ejemplo de celdas H3 (fuente: propia)

Es por todo ello que la motivación de este trabajo consiste en enriquecer los datos de entrada al algoritmo empleado añadiendo nuevos parámetros de estudio basados en celdas H3 con el fin de conseguir mejorar la predicción de tipo de buque. Esto permitirá completar dichos campos en los mensajes AIS para que así el número de buques sospechosos se puede reducir sustancialmente a los que realmente tengan comportamientos extraños e incoherentes.

1.2 Objetivos del TFG

Los propósitos de este trabajo serán segmentados en varios subobjetivos, comenzando con una revisión del trabajo llevado a cabo por AN Gonzalo Rodríguez Casajús durante el curso 2021-2022 en el CUD-ENM [4]. Como se mencionó previamente, este trabajo se centró en el desarrollo de un modelo de predicción basado en los parámetros estáticos de los datos AIS, el cual también demostró una eficacia limitada de los parámetros cinemáticos. Por lo que el trabajo se podría dividir en varias etapas:

- Estado del arte: donde se llevará a cabo un análisis general de *machine learning* y de las celdas H3. Por otro lado, realizará un análisis de los trabajos previos que hayan empleado celdas H3 y su posible impacto en el comportamiento de los barcos.
- Análisis y estudio del código previo definido en [4] .
- Definición de parámetros basados en las celdas H3 que sirvan de entrada al algoritmo de aprendizaje supervisado.
- Comprensión y análisis de los atributos dinámicos basados en la cinemática del barco, y los atributos dinámicos basados en las celdas H3 mediante experimentos conjuntos e independientes.
- Experimentos con todos los tipos de datos y sus combinaciones: estáticos y dinámicos (cinemáticos y celdas H3).
- Análisis y evaluación de los resultados obtenidos.

Con todo ello el objetivo principal del TFG es ver la influencia de las áreas de actividad, mediante la utilización de celdas H3, en los modelos de predicción y conseguir incrementar la calidad de dichas predicciones.

1.3 Organización de la memoria

Tras contextualizar el trabajo y habiendo expuesto los objetivos que se desean cumplir, se pasa a la descripción de la memoria con el fin de mostrar claramente su estructura. El documento quedará dividido en cinco capítulos, la bibliografía y los anexos.

- Capítulo 1: se realiza una contextualización general del trabajo, los motivos de su realización y los principales objetivos marcados.
- Capítulo 2: en este capítulo se desglosará el estado del arte del trabajo. Desde la Inteligencia Artificial como rama de la ciencia hasta los diferentes tipos de algoritmos en base a los tipos de aprendizaje. Por otro lado, se expondrán los diferentes tipos de indexado geoespacial, particularizando en las celdas H3 que son las que se utilizarán en este proyecto.
- Capítulo 3: este capítulo contiene el desarrollo del TFG. Inicialmente se ha expuesto el entorno de trabajo, así como las librerías a utilizar. Después una explicación de los tipos de datos que se van a requeridos y las funciones que se van a aplicar para optimizar el código. Por último, un resumen de los experimentos que se van a realizar.
- Capítulo 4: análisis y explicación de los experimentos realizados.
- Capítulo 5: conclusiones obtenidas y posibles líneas de investigación en el futuro.
- Por último, se adjuntan las referencias bibliográficas y sitios web consultados, así como los anexos en los que se recoge, entra otra información, el código desarrollado.

2 ESTADO DEL ARTE

Este capítulo se centrará en explorar el concepto de Inteligencia Artificial (IA) y el avance reciente en esta tecnología. Se profundizará en los diferentes métodos de Machine Learning: el aprendizaje supervisado, no supervisado y por refuerzo. Se presentarán también los algoritmos de aprendizaje supervisado y no supervisado más comúnmente utilizados para el desarrollo de sistemas inteligentes. Además, se describirá el conocimiento del entorno marítimo y su aplicación en nuestro país, más en particular en la Armada. Por otro lado, se expondrán los diferentes tipos de indexado geoespacial deteniéndonos en las celdas H3, cuya influencia en la detección del tipo de buque tratará de analizar este TFG. Finalmente, se mencionarán algunos proyectos previos relacionados que abarcan objetivos similares al de este Trabajo Fin de Grado.

2.1 Inteligencia Artificial

2.1.1 Definición

Según la RAE se define inteligencia artificial (IA) como *la atribuida a las máquinas capaces de hacer operaciones propias de los seres inteligentes (con capacidad de entender o comprender)* [6].

Realmente definir inteligencia artificial es algo mucho más complejo. El principal problema radica en que la inteligencia es algo que hoy en día todavía no está definido, aunque se basa fundamentalmente en dos vertientes: inteligencia como proceso mental y razonamiento y la segunda como conducta. En la Tabla 2-1, se pueden ver ocho definiciones de la inteligencia artificial. Aunque la primera definición oficial que no recoge esta tabla fue acuñada en 1956, durante la conferencia de Dartmouth por John McCarthy “la inteligencia artificial es la ciencia e ingenio de hacer máquinas inteligentes, especialmente programas de cómputo inteligentes”.

Sistemas que piensan como humanos	Sistemas que piensan racionalmente
"El nuevo y excitante esfuerzo de hacer que los computadores piensen… máquinas con mentes, en el más amplio sentido literal". (Haugeland, 1985) "[La automatización de] actividades que vinculamos con procesos de pensamiento humano, actividades como la toma de decisiones, resolución de problemas, aprendizaje…" (Bellman, 1978)	"El estudio de las facultades mentales mediante el uso de modelos computacionales". (Charniak y McDermott, 1985) "El estudio de los cálculos que hacen posible percibir, razonar y actuar". (Winston, 1992)
Sistemas que actúan como humanos	**Sistemas que actúan racionalmente**
"El arte de desarrollar máquinas con capacidad para realizar funciones que cuando son realizadas por personas requieren inteligencia" (Kurzweil, 1990) "El estudio de cómo lograr que los computadores realicen tareas que, por el momento, los humanos hacemos mejor" (Rich y Knight, 1991)	"La Inteligencia Computacional es el estudio del diseño de agentes inteligentes". (Poole et al., 1998) "IA… está relacionada con conductas inteligentes en artefactos". (Nilsson, 1998)

Tabla 2-1. Algunas definiciones de inteligencia artificial, organizadas en cuatro categorías (fuente: [7])

2.1.2 Contexto histórico

Aristóteles (383-322 a.C.) fue el primero en definir las leyes en las que se basaba la parte racional de la inteligencia, pero no fue hasta 1315 d.C. cuando Ramón Hull planteó la idea de la máquina como objeto capaz de llegar a razonar de forma artificial. A partir de ahí se acuñó que una serie de normas pueden describir la parte racional de la mente, pero faltaría definir la propia mente como concepto físico. Descartes generó la primera discusión donde diferenciaba la mente y la materia, teniendo la primera la capacidad de decidir (libre albedrío).

Fue en la Universidad de Illinois donde se construyó el primer modelo matemático de una neurona del cerebro humano. Warren McCulloch y Walter Pitts partieron de tres fuentes para su desarrollo: conocimientos de la fisiología básica y mecanismo de las neuronas en el cerebro, el análisis de la lógica de Russell y Whitehead y la teoría de computación de Turing [7].

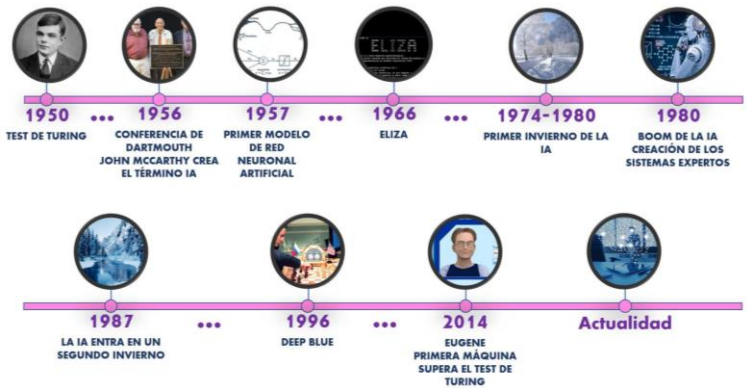


Figura 2-1. Eje cronológico de hitos de la IA (fuente: [4])

Como podemos ver en la Figura 2-1, hablar de Alan Turing sin hablar de IA es impensable. En 1950, en su artículo *Computing Machinery and Intelligence*, desarrolla la prueba de Turing, siendo su fin

principal el que las máquinas que superan dicha prueba pueden ser consideradas como inteligentes. La Figura 2-2 muestra que en la prueba se utilizan tres sujetos A, B y C. El sujeto A es una supuesta máquina inteligente, B un ser humano y el C el interrogador. El fin es conseguir que C determine quién es el humano y quién es la máquina mediante una serie de preguntas. La máquina (A) intentará engañar al interrogador (C) haciéndose pasar por humano [8].

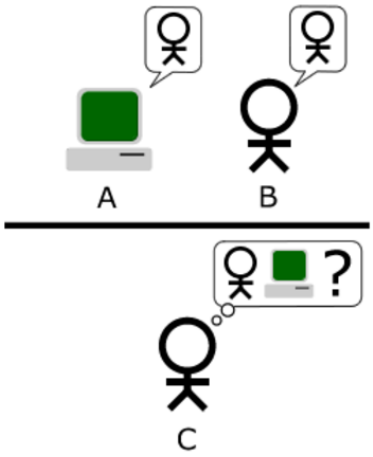


Figura 2-2. Prueba de Turing (fuente: [9])

Como dijo el profesor Geoffrey Jefferson en una de sus conferencias, “hasta que una máquina pueda escribir un soneto o componer un concierto porque sienta los pensamientos y las emociones, y no porque haya una lluvia de símbolos, podría reconocer que la máquina iguala al cerebro, es decir, no sólo escribirlo, sino que sepa que ha hecho” [7]. Si bien es cierto que a lo que concierne como inteligencia artificial no es relevante si la máquina tiene consciencia de lo que está realizando, lo verdaderamente importante es que se comporte de dicha manera. Posteriormente en la conferencia de Dartmouth en 1956, se acuñó el término "inteligencia artificial": John McCarthy, Marvin Minsky, Nathaniel Rochester y Claude Shannon decidieron asignarle este nombre para evitar confundirlo con otras ramas de la tecnología como la cibernética.

En las décadas de 1950 y 1960, el campo de la IA experimentó un gran impulso, fomentado en gran parte por el aumento de la capacidad de procesamiento y el aumento de la inversión gubernamental en investigación en tecnología. Durante este período se desarrollaron algoritmos de aprendizaje automático, sistemas de procesamiento de lenguaje natural y sistemas de razonamiento automatizado. Estuvieron llenos de éxitos, entre los más importantes está la creación de ELIZA (Figura 2-3). Creada por Joseph Weizenbaum, este sistema desarrollado en el laboratorio de inteligencia artificial del MIT en los años 60, era un sistema de procesamiento del lenguaje natural que simulaba una sesión de terapia psicológica mediante el uso de patrones predefinidos y respuestas automatizadas. ELIZA fue un experimento para demostrar la posibilidad de la comunicación humano-máquina.

A principios de los años 70 debido una crisis en las grandes potencias (crisis del petróleo en occidente), la IA sufrió un parón en su desarrollo, el *invierno* de la inteligencia artificial. No fue hasta los años 80 cuando el sector privado retomó el interés en este campo. En particular, se desarrollaron unos programas capaces de almacenar conceptos en el ámbito laboral que ayudaban a resolver las dudas a las empresas sin necesidad de contratar, comúnmente eran conocidos como Sistemas Expertos. De nuevo, la aparición de una nueva crisis y las altas expectativas de los inversores en estos nuevos programas desencadenaron el segundo invierno que duraría hasta finales del siglo XX.

Figura 2-3. ELIZA, 1966 (fuente: [10])

La última etapa se desarrolló desde principios del siglo XXI hasta la actualidad. De nuevo, la mejora económica y los avances tecnológicos permitieron que la IA evolucionase mucho en este período. Los principales logros que se llevaron a cabo fueron, entre otros, el Deep Blue creado por IBM, un programa capaz de aprender por sí mismo, evolucionar y acabar ganando un juego elaborado como es el ajedrez, ganando a Kaspárov (maestro de ajedrez, político y escritor ruso) en 1997 [11].

2.1.3 IA en la actualidad

La inteligencia artificial es una herramienta en constante evolución y en la actualidad se está aplicando en una variedad de campos e industrias. Los avances en el aprendizaje automático y el procesamiento de datos masivos han permitido el desarrollo de algoritmos y sistemas cada vez más sofisticados que pueden realizar tareas que antes requerían inteligencia humana.

En la actualidad, la IA se utiliza en una variedad de aplicaciones, como el análisis de datos, el procesamiento del lenguaje natural, el reconocimiento de imágenes y el reconocimiento de voz. Además, también se utiliza en el campo de la robótica, el control de procesos, y en la medicina para ayudar a los médicos a tomar decisiones.

La IA también está transformando la manera en la que las empresas operan y compiten, mejorando la eficiencia y ayudando a las empresas a tomar decisiones teniendo en cuenta mucha más información. A medida que la tecnología continúa avanzando, se espera que la IA tenga un impacto cada vez mayor en la sociedad y la economía, y se está investigando en áreas como IA general, aprendizaje automático ético y seguridad en IA. Sin embargo, también existen preocupaciones sobre los riesgos y desafíos éticos relacionados con el uso generalizado de la IA [12], por lo que es importante continuar investigando y discutiendo estos temas.

A pesar de eso, es un hecho que en la actualidad las máquinas autónomas cada vez están más presentes. Desde máquinas que sustituyen a humanos como pueden ser las máquinas de autoservicio en restaurantes de comida rápida, hasta sistemas como Alexa o Siri (Figura 2-4) que forman parte del día a día de muchas personas. Por último, en noviembre de 2022 apareció *ChatGPT* una inteligencia artificial desarrollada por *OpenAI*. Dicho programa fue entrenado para producir textos coherentes y originales en respuesta a una pregunta. Se basa en la técnica de transformación y utiliza una red neuronal para producirlos. La creación de *ChatGPT* ha mejorado la capacidad de los modelos de lenguaje para comprender y producir textos humanos y se usa en aplicaciones como asistentes virtuales y sistemas de traducción automática. Sin embargo, hay preocupaciones éticas sobre su uso en la producción de falsificaciones. Es importante considerar las implicaciones éticas y sociales de su uso [13].

Figura 2-4. Siri (fuente: [14])

2.2 Machine Learning

El concepto de Machine Learning (ML) nació poco después que el de IA. Fue Arhur L. Samuel quien en 1959 acuñó este término en su investigación con el juego de las damas [15]. El fin principal era conseguir, mediante el uso de datos y algoritmos, que las máquinas consiguiesen aprender de la misma forma en la que los seres humanos lo hacen. Fue así cuando en 1962 Robert Nealey, el autoproclamado maestro de las damas, perdió contra un ordenador IBM 7094.

El concepto básico de aprendizaje automático en ciencia de datos implica el aprendizaje estadístico y los métodos de optimización, que permiten, a los ordenadores analizar datos e identificar patrones. Estas técnicas aprovechan el procesamiento de datos para clasificar tendencias históricas y poder predecir futuros modelos. El algoritmo básico de ML cuenta con tres componentes fundamentales [16]:

- Proceso de decisión: en base a un conjunto de datos de entrada (que pueden estar clasificados o no) el algoritmo estima un patrón de los datos.
- Función error: en caso de tener datos conocidos, se puede realizar una comparación de cuan buena es nuestra estimación inicial.
- Proceso de optimización de modelos: una vez realizada la evaluación de los datos, el proceso repetirá la estimación del patrón con el fin de alcanzar el umbral de precisión deseado.

2.2.1 Machine Learning, Deep Learning, Redes neuronales

En la actualidad Machine Learning, Deep Learning y Redes neuronales tienden a utilizarse indistintamente, por lo que es interesante señalar las principales diferencias. Todos ellos son subcampos que fueron naciendo a partir de la IA como se puede ver en la Figura 2-5. No obstante, el aprendizaje automático engloba a los otros dos y las redes neuronales constituyen la columna vertebral de los algoritmos de aprendizaje profundo.

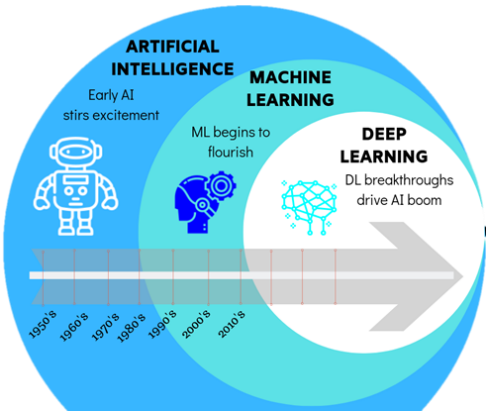


Figura 2-5. Esquema IA vs ML vs DL (fuente: [17])

Machine Learning es una rama de la inteligencia artificial que se enfoca en el desarrollo de algoritmos y técnicas que permiten a las máquinas aprender de forma autónoma a partir de datos. Los expertos humanos son los encargados de determinar las diferencias de los datos de entrada por lo que es un proceso más lento.

Deep Learning es un subconjunto del ML que se enfoca en el uso de redes neuronales profundas para analizar datos complejos. Estas redes neuronales son capaces de aprender patrones y relaciones en los datos de manera automática, a diferencia del Machine Learning clásico que se basa en la intervención humana. Lex Fridman lo denominó como un "aprendizaje automático escalable" por su capacidad de utilizar cada vez un número mayor de datos [18].

Las redes neuronales son una técnica de Machine Learning y Deep Learning que se basa en la simulación de la estructura y funcionamiento de las redes de neuronas en el cerebro humano. Estas redes están compuestas por capas de nodos (neuronas) que se conectan entre sí y se activan mediante una serie de procesos matemáticos, permitiendo que la red aprenda y se adapte a nuevos datos. Estas redes son utilizadas en aplicaciones como el reconocimiento de patrones, la clasificación de imágenes y el análisis de datos.

2.2.2 Aprendizaje supervisado

El aprendizaje supervisado es un tipo de aprendizaje automático en el que se proporciona al sistema un conjunto de datos de entrenamiento etiquetados previamente, y se le pide que aprenda a realizar una tarea específica a partir de estos datos. Por ejemplo, como se puede ver en la Figura 2-6, un proceso de aprendizaje supervisado sería que el programa fuese capaz de clasificar perros y gatos a partir de las imágenes de dichos animales. Para ello los datos de entrenamiento tendrán que estar correctamente clasificados, y mediante estos datos históricos el algoritmo aprenderá a diferenciarlos y a aplicarlo sobre los diferentes datos de entrada [19].

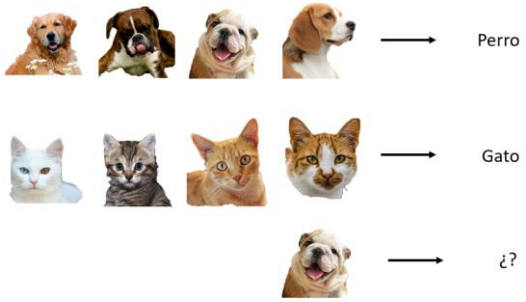


Figura 2-6. Ejemplo de aprendizaje supervisado, clasificación (fuente: propia)

Podemos distinguir dos tipos de aprendizaje supervisado:

1. Clasificación: a partir del cual el sistema aprende a diferenciar entre varios casos. Por ejemplo, si un correo electrónico es spam o no.
2. Regresión: en función de unos datos de entrada determinar un valor de salida. Por ejemplo, determinar el valor de una casa en función de su ubicación, tamaño y antigüedad.

En el proceso de entrenamiento es importante dividir los datos en dos conjuntos: un conjunto de entrenamiento y un conjunto de prueba. El conjunto de entrenamiento se utiliza para entrenar el modelo, mientras que el conjunto de prueba se utiliza para evaluar la precisión del modelo y, de esta manera, ir depurando los errores hasta llegar al umbral de precisión deseado.

2.2.3 Aprendizaje no supervisado

El aprendizaje no supervisado se diferencia en que el sistema es capaz de aprender por sí mismo sin necesidad de ser guiado por un conjunto de etiquetas de clasificación. En lugar de eso, se le da al sistema un conjunto de datos no etiquetados y se le permite explorar y encontrar patrones en los datos por sí mismo. En la Figura 2-7 se puede observar un grupo de datos etiquetados (grupo de entrenamiento) y otro de datos no etiquetados (test).

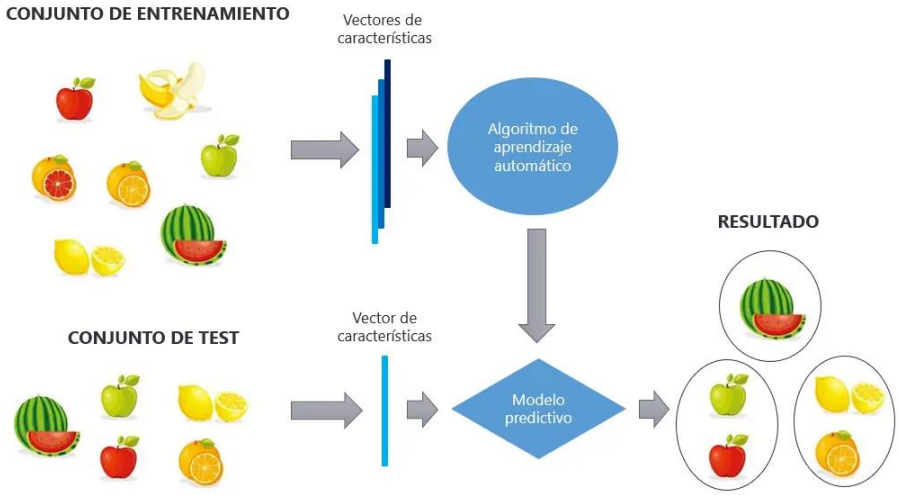


Figura 2-7. Ejemplo de aprendizaje no supervisado (fuente: [20])

Podemos encontrar diferentes tipos:

- *Clustering* (agrupamiento): el sistema recibe un conjunto de datos y se le pide que los agrupa en diferentes conjuntos. Este tipo de aprendizaje se utiliza en el análisis de segmentación de clientes, en el cual se busca agrupar a los clientes en función de unas características determinadas (demográficas, comportamientos de compra…) [21].
- La reducción de dimensionalidad: se utiliza para simplificar y visualizar los datos de un conjunto de datos de alta dimensión. El algoritmo busca encontrar un conjunto de características principales que resuman la información contenida en el conjunto de datos original, lo cual puede ser útil para la visualización y el análisis de estos. Se puede hacer utilizando gran variedad de algoritmos, como la técnica de análisis de componentes principales (PCA) o la técnica de análisis de componentes independientes (ICA). Por ejemplo, en el ámbito de las imágenes es muy útil, consiguiendo reducir su tamaño (número de bytes) sin que sea apreciable para el ojo humano [22].
- Asociación: busca encontrar patrones de relación entre diferentes variables en un conjunto de datos. Por ejemplo, se puede utilizar para analizar las compras de un cliente y determinar qué productos suelen comprarse juntos.

- Detección de anomalías: es un proceso en el que mediante algoritmos de aprendizaje no supervisado se entrena un modelo con datos conocidos como "normales" y se utiliza para identificar datos atípicos o "anómalos" en un conjunto de datos futuros. Es una técnica utilizada para detectar eventos inusuales o patrones anómalos en los datos, como transacciones fraudulentas o fallos en un sistema.

2.2.4 Aprendizaje semisupervisado

El aprendizaje semisupervisado es una técnica de aprendizaje automático en la que se entrena un modelo con un conjunto de datos parcialmente etiquetados. En comparación con el aprendizaje supervisado, donde se proporciona un conjunto completo de datos etiquetados, en el aprendizaje semisupervisado solo se proporciona un subconjunto de los datos con etiquetas. El objetivo es utilizar estos datos etiquetados para inferir las etiquetas de los que no las tienen.

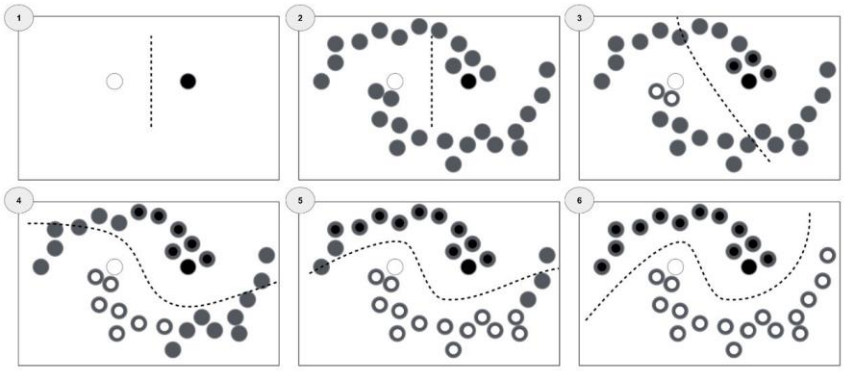


Figura 2-8. Ajuste de la frontera de decisión en aprendizaje semisupervisado (fuente: [23])

Hay varias razones por las que se utiliza el aprendizaje semisupervisado. En primer lugar, porque el tiempo que se consume etiquetando todos los datos es elevado. Además, en algunos casos, puede ser difícil obtener datos etiquetados. El aprendizaje semisupervisado permite el uso de un conjunto de datos no etiquetados para mejorar el rendimiento del modelo, lo que puede ser útil en aplicaciones como el procesamiento del lenguaje natural o la visión por computador.

Hay varios métodos para el aprendizaje semisupervisado, un ejemplo sería el que se puede ver en la Figura 2-8. Además, hay otros tipos como el aprendizaje por transferencia, el aprendizaje por etiquetado automático, la inferencia de etiquetas y el aprendizaje de etiquetas no observadas.

2.2.5 Aprendizaje reforzado

El aprendizaje reforzado es una subcategoría del aprendizaje automático en la que un agente aprende a tomar acciones en un entorno para maximizar una recompensa. El agente aprende a través de la experiencia, mediante la realización de acciones y la observación de las consecuencias de esas acciones. El aprendizaje reforzado se utiliza ampliamente en problemas de toma de decisiones y control, como los sistemas de control de tráfico aéreo, los robots industriales y los juegos de computadora.

En el aprendizaje reforzado, el agente interactúa con el entorno mediante la selección de acciones. Cada acción tiene una recompensa asociada, que puede ser positiva o negativa. El objetivo del agente es maximizar la recompensa acumulada a lo largo del tiempo. Esto se logra mediante la selección de acciones que se espera que generen una recompensa mayor.

Para entender cómo funciona el aprendizaje reforzado, es importante conocer algunos términos clave como agente, entorno, acción y estado [24]. También los podemos ver en la Figura 2-9:

- Agente es el programa que se entrena y que luego será el encargado de tomar decisiones.
- Entorno es el ambiente en el que opera el agente.
- Acciones son los movimientos realizados por el agente en el entorno.

- Estado es la situación en la que se encuentra el entorno cada vez que cambia debido a las acciones del agente.
- Recompensa es una forma de retroalimentación que se proporciona al agente para evaluar la acción que ha tomado. Es la encargada de informar al algoritmo si la decisión tomada fue acertada o equivocada. La recompensa puede ser positiva o negativa, lo que significa que puede ser una recompensa por una acción correcta o un castigo por una acción equivocada.

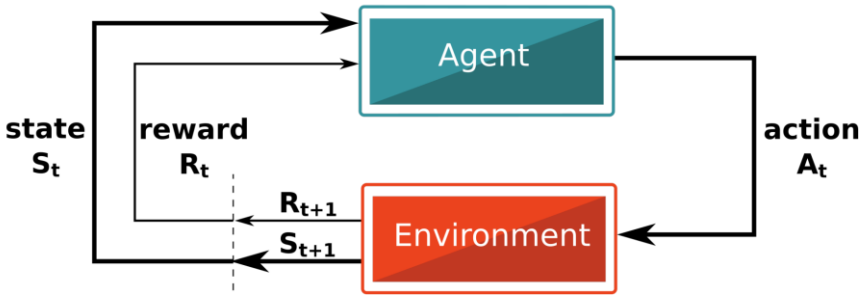


Figura 2-9. Proceso de aprendizaje por refuerzo (fuente: [25])

En la actualidad, los desarrollos más recientes en el aprendizaje reforzado se enfocan en abordar problemas de secuencia a secuencia (seq2seq) mediante el uso de mecanismos de atención y la capacidad de entrenamiento paralelo proporcionada por los Transformers (modelo de red neuronal). Un ejemplo de esto es Gato (Figura 2-10), una IA generalista diseñada con estas características, que puede completar oraciones, jugar videojuegos, manipular objetos con un brazo robótico y funcionar como un chatbot, todo con un solo modelo y sin necesidad de volver a entrenarlo para cada tarea.

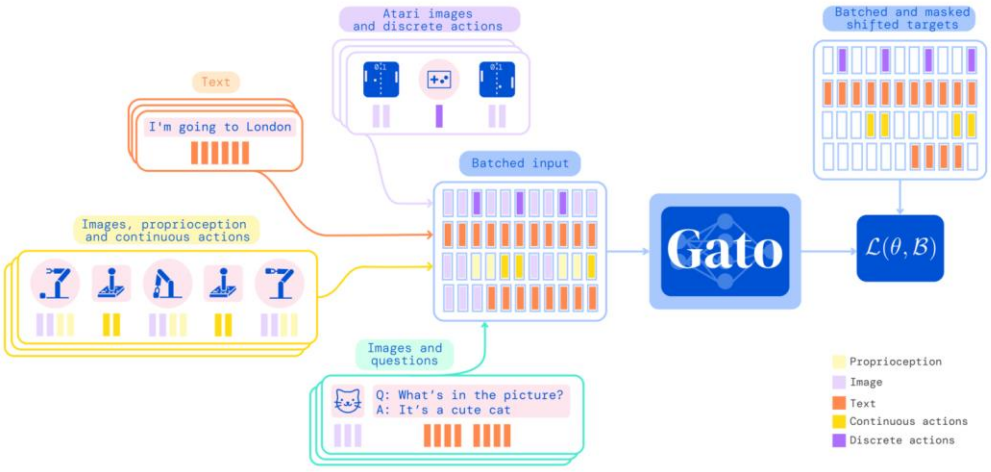


Figura 2-10. Gato: modelo de secuencia de aprendizaje reforzado (fuente: [26])

2.3 Clasificación de algoritmos

2.3.1 Algoritmos de aprendizaje supervisado

Habiendo explicado ya los principios básicos del aprendizaje supervisado (2.2.2), en este apartado se realizará un breve resumen de los principales algoritmos [27].

2.3.1.1 Árboles de decisión

Un árbol de decisión es un algoritmo de aprendizaje supervisado no paramétrico. Comúnmente son utilizados para tareas tanto de regresión como de clasificación. Las partes de un árbol de decisión (Figura 2-11) son las siguientes:

- Nodo raíz: es el punto principal a partir del cual se forma el resto del árbol.
- Ramas: son los posibles caminos que unen a los diferentes nodos de distinto nivel
- Nodos internos: o nodos de decisión, son los "cruces" donde se elige cual es el camino que se va a seguir, realizando una evaluación para formar subconjuntos homogéneos.
- Nodos hoja: son los posibles resultados del conjunto de datos.

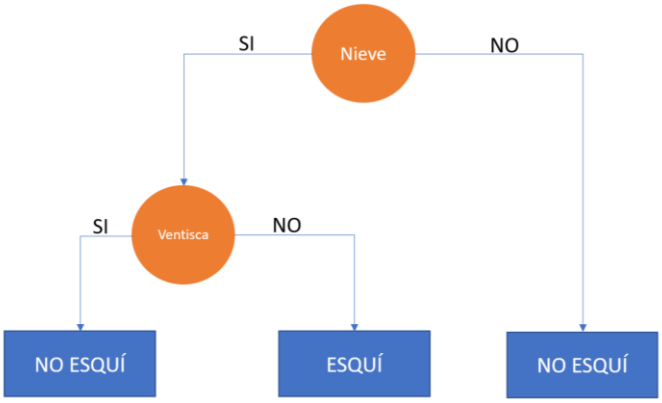


Figura 2-11. Ejemplo de árbol de decisión (fuente: propia)

El algoritmo *Hunt* es una de las bases para muchos algoritmos de árboles de decisión. Entre los que destacan: ID3, desarrollado por Ross Quinlan, donde se utiliza la entropía y la ganancia de información para evaluar las divisiones de los posibles nodos hoja. *CART*, introducido por Leo Breiman, usa la impureza de Gini (que mide la tasa de clasificación incorrecta buscando el valor más bajo posible) para identificar el mejor atributo para la división. En definitiva, para tener en cuenta el mejor atributo en cada nodo hay que tener en cuenta los siguientes tres parámetros [28]:

- Entropía: este concepto mide la impureza de la muestra, se mide entre 0 y 1. Por ejemplo, si todas las muestras en el conjunto de datos pertenecen a la misma clase, la entropía seria de valor 0, por el contrario, si son todas de diferentes clases, el valor sería 1.
- Ganancia de información: representa la diferencia de entropías antes y después de la división en un atributo determinado.
- Impureza de Gini: es un valor que va de 0 a 1 y se utiliza para medir la homogeneidad de un conjunto de datos. Se basa en la probabilidad de elegir dos elementos al azar del conjunto y que sean del mismo tipo; si son iguales, el índice es 1. Se prefieren atributos con un índice de Gini bajo en lugar de uno alto, para que estén más cerca del nodo raíz. Sin embargo, este método solo es aplicable a decisiones binarias limitando mucho su utilización.

$$Impureza\ de\ Gini = 1 - \sum_i (P_i)^2$$

2.3.1.2 Clasificación de Naïve Bayes

En términos generales, los modelos de Naïve Bayes son un tipo de algoritmos que se basan en el teorema de Bayes, Figura 2-12, una técnica estadística de clasificación.

Estos algoritmos son conocidos como "naïve" o "inocentes", ya que suponen que las variables predictoras son independientes entre sí, es decir, la presencia de una característica en los datos no está relacionada con la presencia de otras características. Son fáciles de construir y ofrecen un buen rendimiento debido a su simplicidad. Esto se logra al proporcionar una forma de calcular la probabilidad posterior de que ocurra un evento, teniendo en cuenta las probabilidades previas de otros eventos.

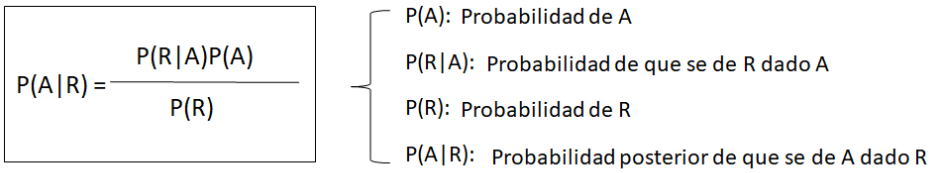


Figura 2-12. Teorema de Bayes (fuente: [29])

El algoritmo de Naïve Bayes es una técnica de clasificación que ofrece una forma eficiente y rápida de predecir clases en problemas binarios y multiclase. La principal ventaja de este algoritmo es que, en casos donde se puede suponer independencia entre variables predictoras, su desempeño es mejor que el de otros modelos de clasificación, incluso con menos datos de entrenamiento. Además, la separación de las distribuciones de características condicionales por clase significa que cada distribución se puede estimar por separado, lo que ayuda a mejorar el rendimiento en problemas de dimensionalidad.

Sin embargo, Naïve Bayes también tiene algunas debilidades. Aunque es un buen clasificador, sus probabilidades no deben ser tomadas demasiado en serio ya que no son muy precisas. Además, la presunción de independencia puede ser poco realista y el modelo no será útil en casos en que haya características en el conjunto de prueba que no se hayan observado en el conjunto de entrenamiento. Una técnica común para superar este problema es el suavizado, por ejemplo, con la estimación de Laplace [29].

2.3.1.3 Regresión lineal

El objetivo del análisis de regresión es encontrar una relación entre una serie de variables y un resultado continuo. Se trata de una rama del aprendizaje automático (ML) supervisado que busca establecer un modelo que permita predecir resultados cuantitativos, como el precio de una propiedad o el tiempo que alguien pasará viendo un vídeo.

Se trata de encontrar la línea que mejor se ajuste a los puntos de datos para representar la relación entre una característica independiente y su resultado correspondiente. El algoritmo irá moviendo y recalculando la línea para minimizar los errores de predicción y encontrar la mejor aproximación a los datos.

$$y = mx + b$$

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

El algoritmo busca representar los datos de entrada (X) estableciendo un valor de salida (Y) mediante la ecuación de la recta (Figura 2-13) que represente con el menor error posible el conjunto de datos. Se busca que la recta se acerque cada vez más a los puntos. Para lograr esto, el algoritmo usará un factor conocido como "tasa de aprendizaje". Esta tasa de aprendizaje es un número que multiplica los parámetros de la recta para realizar pequeñas aproximaciones hacia los puntos [30].

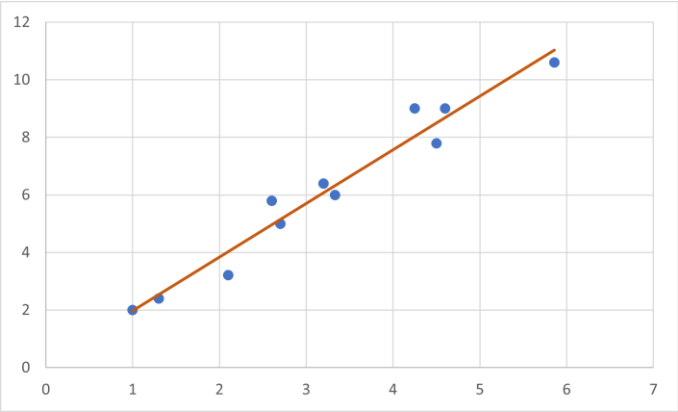


Figura 2-13. Recta de regresión lineal (fuente: propia)

Dentro de la regresión lineal existe un caso particular que es la regresión polinomial como se puede ver en la Figura 2-14. Cuando la salida que se busca predecir está relacionada con más de una variable, se puede utilizar un modelo más complejo que tenga en cuenta estas dimensiones adicionales. La inclusión de más variables relevantes puede mejorar la precisión de las predicciones.

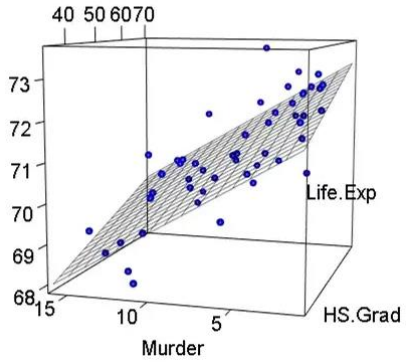


Figura 2-14. Regresión polinomial (fuente: [30])

2.3.1.4 Regresión logística

Es una técnica que se utiliza tanto para hacer predicciones sobre una variable cuantitativa, como para clasificar una muestra en categorías o grupos. El resultado final es una respuesta binaria, es decir, un sí o no en la predicción o clasificación. Sin embargo, en niveles más avanzados, puede ser multinomial (tres o más categorías sin un orden definido) u ordinal (tres o más categorías con un orden definido). En cualquier caso, la respuesta que se busca es una identificación de pertenencia a una categoría específica [31].

2.3.1.5 Vectores de soporte (SVM)

El SVM (*Support Vector Machine*) es un algoritmo de clasificación que trabaja creando una barrera virtual, llamada hiperplano de separación, que divide los datos de entrenamiento etiquetados en dos categorías. Esta barrera se ubica de tal manera que maximiza la distancia entre los puntos más cercanos de ambas categorías, como en la Figura 2-15, conocidos como vectores de soporte. Cualquier punto nuevo que entre en el sistema será asignado a una categoría dependiendo de en qué lado del hiperplano caiga [32].

En este algoritmo se utiliza una técnica llamada *kernel* para separar clases de datos. Esta técnica transforma los datos de entrada en un espacio de mayor dimensión y, a través del uso de funciones llamadas núcleos, convierte un problema de separación no lineal en un problema linealmente separable. De esta manera, el algoritmo puede clasificar los datos en base a las etiquetas o resultados previamente definidos.

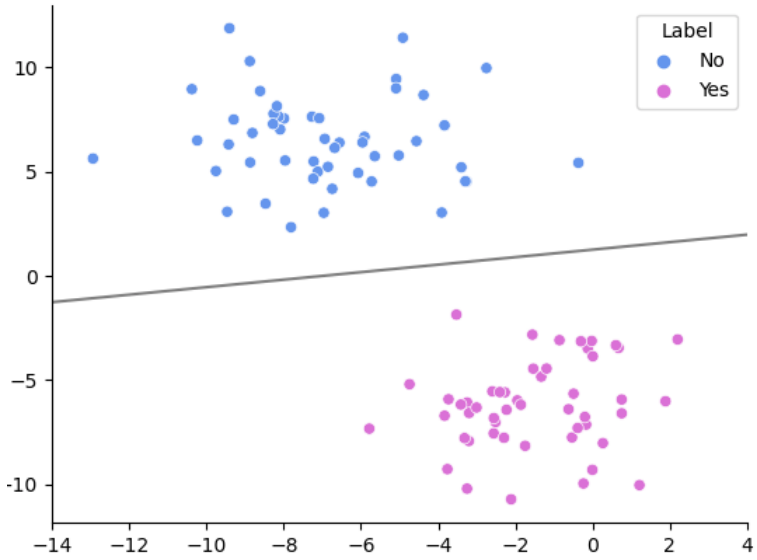


Figura 2-15. División de datos en base a un hiperplano (fuente: [33])

2.3.1.6 K vecinos más cercanos (kNN)

El *kNN* (*k-Nearest Neighbors*) es un algoritmo de aprendizaje automático supervisado que utiliza la cercanía para predecir a qué grupo pertenece dato en particular. Es una técnica no paramétrica que se aplica a tareas de clasificación y regresión, pero más comúnmente se utiliza en clasificación. Funciona asumiendo que los puntos similares están cerca entre sí. A continuación, se puede observar un ejemplo en la Figura 2-19 donde *k*=3.

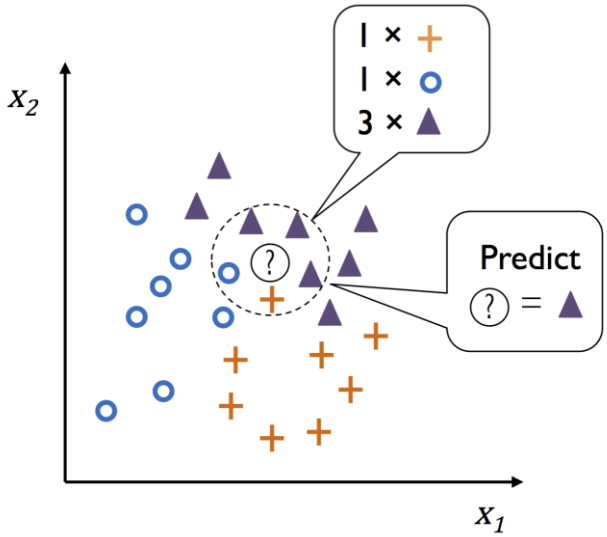


Figura 2-16. Algoritmo *kNN* para problema con 3 clases y k=5 (fuente: [34])

Para calcular la distancia entre los diferentes puntos se pueden utilizar diferentes métodos de medida [35]:

- Distancia euclidiana: es la distancia más comúnmente usada. La medida vectorial queda limitada a valores reales. La fórmula permite calcular la línea recta entre dos puntos, siendo el punto de consulta uno de ellos.

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}$$

- Distancia *Manhattan*: mide la diferencia entre dos puntos en términos de valor absoluto. Se le llama así debido a que se parece a la forma en la que un taxista se mueve en las calles de una ciudad, y es una forma visual de comprender cómo se puede llegar de un punto a otro.

$$d(x,y) = \left(\sum_{i=1}^{m} |x_i - y_i|\right)$$

- Distancia *Minkowski:* esta medida es una versión general de la distancia euclidiana y *Manhattan*. El parámetro *p* en la ecuación permite variar la forma en que se mide la distancia, dando la posibilidad de crear métricas personalizadas. Cuando *p* es igual a dos, se representa la distancia euclidiana y cuando es igual a uno, se representa la distancia de *Manhattan*.

$$(x,y) = \sqrt[p]{\left(\sum_{i=1}^{m} |x_i - y_i|\right)}$$

- Distancia *Hamming*: se utiliza comúnmente con vectores de tipo booleano o de cadena y se basa en contar los puntos donde los vectores no son iguales. Debido a esto, también se le conoce como la métrica de la superposición. Se representa matemáticamente mediante la siguiente fórmula:

$$D_H = \left(\sum_{i=1}^{k} |x_i - y_i|\right) \;||\; x = y \rightarrow D = 0 \;;\; x \neq y \rightarrow D \neq 1$$

2.3.1.7 Random Forest

Random Forest o bosque aleatorio es un algoritmo diseñado para superar las limitaciones de los árboles de decisión (2.3.1.1). A menudo, los árboles de decisión pueden ser muy precisos con los datos de entrenamiento, pero pierden precisión cuando se aplican a otros datos debido a un problema conocido como sobreajuste. *Random Forest* se creó para resolver estos problemas y mejorar la precisión al reducir el sobreajuste [36]. El sobreajuste ocurre cuando un modelo se vuelve demasiado específico para los datos de entrenamiento y no es capaz de generalizar patrones a otros datos nuevos. Este es un método de aprendizaje automático muy valorado debido a sus muchas ventajas en comparación con otros algoritmos. Es fácil de comprender, robusto y suele proporcionar resultados precisos.

Este algoritmo implica la creación de *n* árboles de decisión que son generados aleatoriamente sin poda. Estos árboles se construyen a partir de los datos del *dataset* que están divididos en datos de entrenamiento y datos de prueba, que son seleccionados de forma aleatoria. El proceso de crear árboles a partir de datos aleatorios se llama *bagging*. El usuario también selecciona las características que el modelo debe utilizar para hacer divisiones. Para realizar tareas de clasificación o regresión con nuevos datos, se introduce el dato de entrada en cada uno de los árboles generados y la predicción será el valor más votado (clasificación) o el valor promedio de salida de cada árbol (regresión).

La Figura 2-17 muestra cómo un objeto se propaga a través de diferentes tipos de árboles de *Random Forest* (RF) [37]. En un árbol RF regular (izquierda), el objeto sigue un solo camino basado en criterios binarios. En un RF ideal (centro), el objeto se dispersa a través de todas las ramas con las probabilidades calculadas y almacenadas en el camino. En la imagen de la derecha (aproximación RF) muestra la

influencia de un parámetro umbral de probabilidad provocando que se descarten las ramas con probabilidades bajas (las marcadas con X), reduciendo así el tiempo de ejecución.

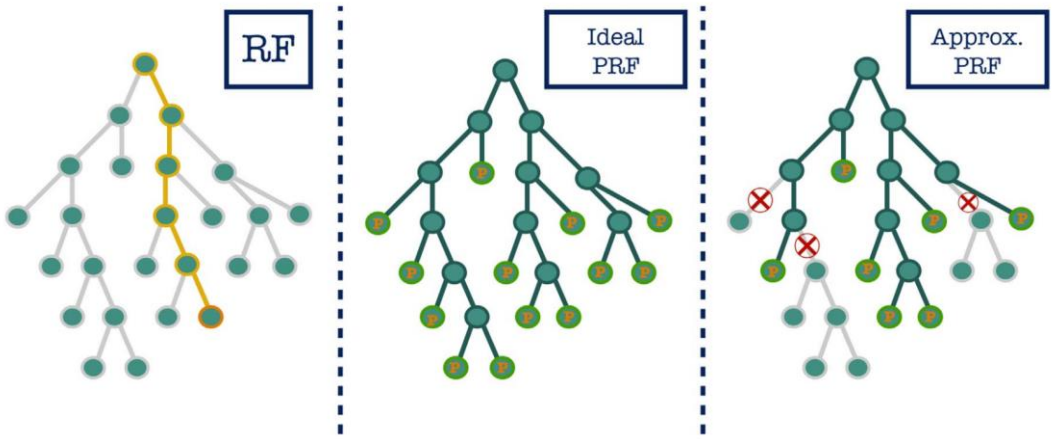


Figura 2-17. Ejemplo de *Random Forest* (fuente: [37])

2.3.2 Algoritmos de aprendizaje no supervisado

En el apartado 2.2.3 se explicaron los principios del aprendizaje no supervisado. En este apartado se explicarán los algoritmos más importantes que se pueden encontrar.

2.3.2.1 Agrupación o clustering

La agrupación consiste recibir un conjunto de datos no etiquetados y dividirlos en diferentes grupos (*clusters*) en función de las distancias entre los datos. La principal diferencia con la clasificación en el aprendizaje supervisado es que este algoritmo busca agrupar los datos de una forma natural sin que dichos datos estén etiquetados [38]. Dentro de los agrupamientos podemos encontrar diferentes tipos [39]:

2.3.2.1.1 Clusterización jerárquica

La clusterización jerárquica es una técnica de agrupamiento que clasifica los datos según su similitud. El proceso consiste en calcular la distancia entre todos los datos, y luego agrupar los más cercanos, repitiendo este proceso tantas veces como sea necesario hasta formar el número de *clusters* deseados o hasta alcanzar una distancia máxima establecida por el usuario.

Este proceso se puede visualizar en un dendograma, una representación gráfica que muestra cómo los datos se van uniendo y la distancia que los separa. El usuario puede detener el agrupamiento cuando se han obtenido los grupos deseados (división) o trazar una línea horizontal (aglomeración) en el dendograma en la distancia máxima deseada para obtener los *clusters* [40]. En la Figura 2-18 se puede ver un ejemplo en el que se divide un conjunto de datos en tres *clusters* en base a lo explicado anteriormente.

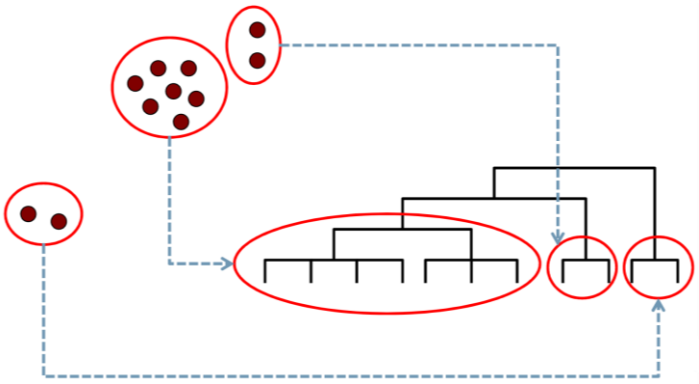


Figura 2-18. Clusters = 3 (fuente: [41])

2.3.2.1.2 K-medias

K-medias o *k-means* es un algoritmo que agrupa objetos en base a sus similitudes. La idea detrás de este algoritmo es dividir los objetos en *k* grupos, donde *k* es el número de *clusters* especificado previamente. El proceso se lleva a cabo minimizando la distancia entre cada objeto y el centro de su grupo. Las fases del proceso son tres [42]:

1. Inicialización: se eligen *k* centros (Figura 2-19) en el espacio de datos, generalmente se suelen elegir de manera aleatoria.

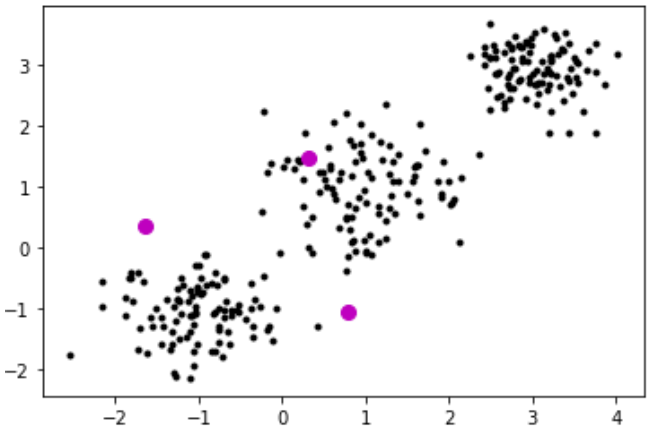


Figura 2-19. Fase 1 *k*-medias, *k*=3 (fuente: [42])

2. Asignación de objetos a los centroides: cada objeto es asignado al centro más cercano.
3. Actualización de centroides: la posición de cada centro es actualizada buscando reducir el promedio de los objetos en su grupo como se puede ver en la Figura 2-20.

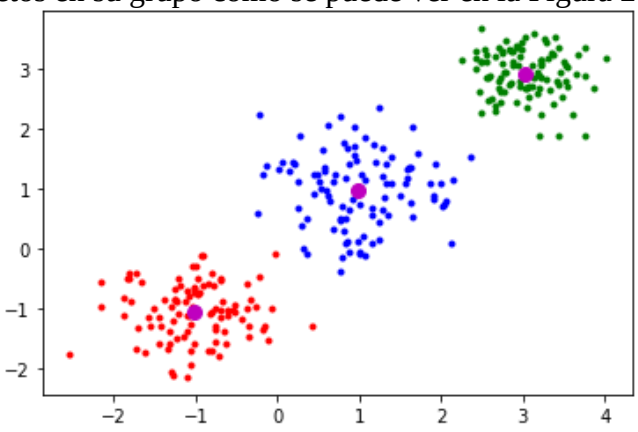


Figura 2-20. Actualización final centroide (fuente: [42])

Los pasos 2 y 3 se repiten tantas veces como sea necesario hasta conseguir que los centros sean fijos o no superen un umbral determinado anteriormente. Por lo que se puede concluir que *k*-medias es un problema de optimización en el que se busca minimizar la distancia entre cada dato y su centro.

2.3.2.1.3 DBSCAN

El algoritmo DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) se basa en el análisis de las densidades de los datos. Se marca un umbral de densidad establecido por la distancia de dos datos y se especifica el número de datos necesario para superar dicho umbral [43].

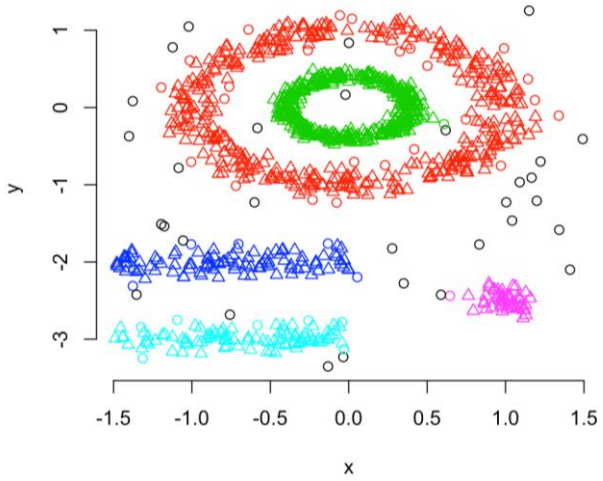


Figura 2-21. Algoritmo DBSCAN (fuente: [44])

Las principales ventajas de este algoritmo de clusterización son que no requiere especificar el número de *clusters* necesarios y la elevada velocidad en términos de implementación. En la Figura 2-21 podemos observar un ejemplo donde se dividen un conjunto de datos en cinco *clusters*, También es cierto que este algoritmo queda limitado a la hora de manejar grandes cantidades de datos y tiene la necesidad de tener definida la distancia entre datos.

2.3.3 Reducción de la dimensionalidad

Como ya se explicó en el apartado 2.2.3, la reducción de la dimensionalidad es un algoritmo que busca simplificar el conjunto de datos con el fin de obtener una mayor eficiencia de estos. Los dos algoritmos más conocidos son los siguientes:

2.3.3.1.1 PCA

El Análisis de Componentes Principales o PCA consiste en un algoritmo para simplificar la cantidad de variables de un conjunto de datos mientras se mantiene tanta información como sea posible. Este proceso involucra la transformación de un gran número de variables en un conjunto más pequeño.

La reducción de la dimensionalidad produce una disminución en la precisión, pero la idea del PCA es que se puede sacrificar un poco de precisión para obtener una representación más simple y manejable del conjunto de datos. Al tener un conjunto de datos de menor dimensionalidad, es más fácil explorar y visualizarlos, y también es más rápido y eficiente para los algoritmos de aprendizaje automático que no tienen que lidiar con variables innecesarias [45].

2.3.3.1.2 ICA

El Análisis de Componentes Independientes (ICA) es un algoritmo que se utiliza para separar una combinación de varias señales en sus componentes individuales. La idea detrás de ICA es que cada componente individual sea lo más independiente posible entre sí. Esta independencia se puede lograr mediante la eliminación de la correlación entre los componentes.

El objetivo es aprovechar las propiedades estadísticas de los datos para separar la señal en componentes distintos. ICA se utiliza en una amplia variedad de aplicaciones, desde la separación de música y voces en audio hasta la separación de señales cerebrales en neurociencia [46].

Dicha técnica es completamente autónoma y utiliza la información contenida en los datos para separar la señal. Este enfoque no supervisado lo convierte en una herramienta valiosa para explorar y comprender la estructura de los datos, especialmente cuando se desconoce la cantidad o la naturaleza de las componentes individuales.

2.4 Conocimiento del entorno marítimo

Las actividades en la mar que ponen en peligro nuestra seguridad coexisten con otras que son absolutamente necesarias y legales. El desafío es supervisar integralmente y monitorear todas las actividades para poder distinguirlas. Por lo tanto, para que la seguridad marítima sea efectiva, se requiere de un conocimiento del entorno marítimo (CEM). Este conocimiento se obtiene a través de la presencia naval y la vigilancia marítima y es un elemento clave para actuar contra las actividades que representan una amenaza en la mar [47].

Para obtener el CEM, se requiere una arquitectura que soporte la obtención, fusión, correlación, análisis y diseminación de una gran cantidad de información sobre barcos, personas, cargamentos, infraestructuras y zonas de interés. Hay muchos sistemas y servicios de información, tanto civiles como militares, todos ellos pueden contribuir significativamente a mejorar el conocimiento del entorno marítimo si comparten información. Para lograr esto, se necesita un enfoque integral que solo puede conseguirse a través de la colaboración de toda la comunidad marítima, tanto a nivel nacional como internacional.

En comparación con épocas previas, el alcance del ámbito marítimo ya no se limita a la división del mar tradicional como se puede ver en la Figura 2-22: aguas territoriales, zona contigua y la Zona Económica Exclusiva (ZEE), sino que ha evolucionado hacia una comprensión más amplia. La zona marítima de interés ha aumentado y ahora se centra en dos factores: uno geográfico, en donde se encuentren los intereses, y otro temporal, en cada momento específico. Por lo tanto, el concepto de zona marítima de interés nacional es cambiante y muy amplio.

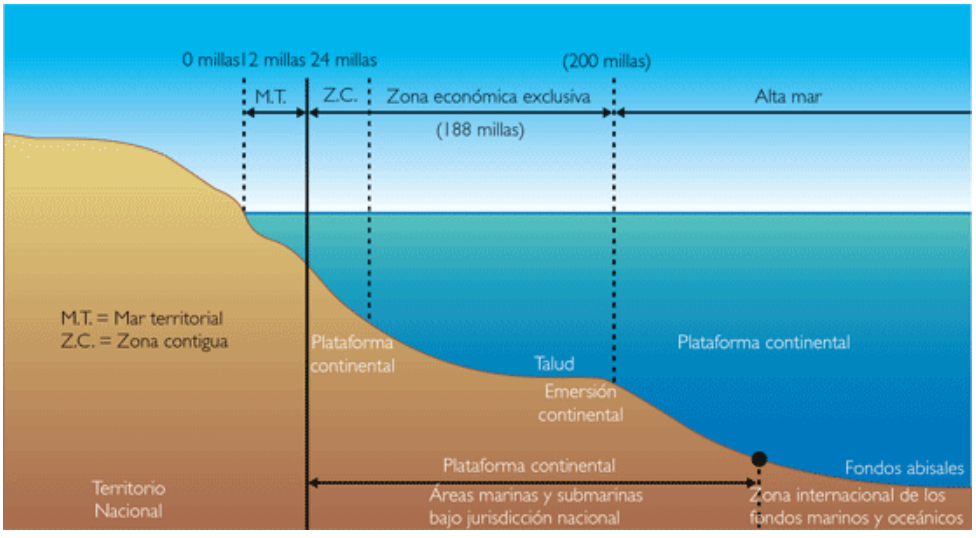


Figura 2-22. División del mar (fuente: [48])

Además, los métodos disponibles para entender el ambiente marino ya no se limitan a los recursos y herramientas provistos por los barcos y otros equipos militares. La globalización y la tecnología moderna han abierto la posibilidad de vigilar, actuar y colaborar con otras organizaciones en regiones lejanas y amplias del país, cada una con sus propias capacidades diseñadas para satisfacer sus

necesidades [49]. En particular el organismo encargado de realizar estas tareas en España es el Centro de Operaciones y Vigilancia de Acción Marítima (COVAM).

2.4.1 Centro de Operaciones y Vigilancia de Acción Marítima (COVAM)

El Centro de Operaciones y Vigilancia de Acción Marítima (COVAM) es la institución española perteneciente a la Armada encargada de controlar el conocimiento del entorno marítimo en España. Fue establecido con la intención de apoyar a los buques de la FAM (Fuerza de Acción Marítima) en la realización de sus misiones. Es desde su ubicación central, en Cartagena, desde donde se puede llevar a cabo una vigilancia marítima continua a través del uso de diversos recursos tanto técnicos como humanos, lo que se refleja en las llamadas tareas de vigilancia. Con esto se busca garantizar la seguridad de los barcos que navegan en la zona de influencia, ayudando así a mantener una seguridad marítima en los espacios marítimos de interés [50]. Además, el COVAM busca colaborar y complementar las actividades de las distintas administraciones públicas del Estado y otros organismos internacionales a través de acuerdos interministeriales. Esto se implementa en los espacios marítimos que son de interés permanente, como son nuestras aguas jurisdiccionales, o temporalmente considerados de interés nacional, como el Índico y el Golfo de Guinea (Figura 2-23).

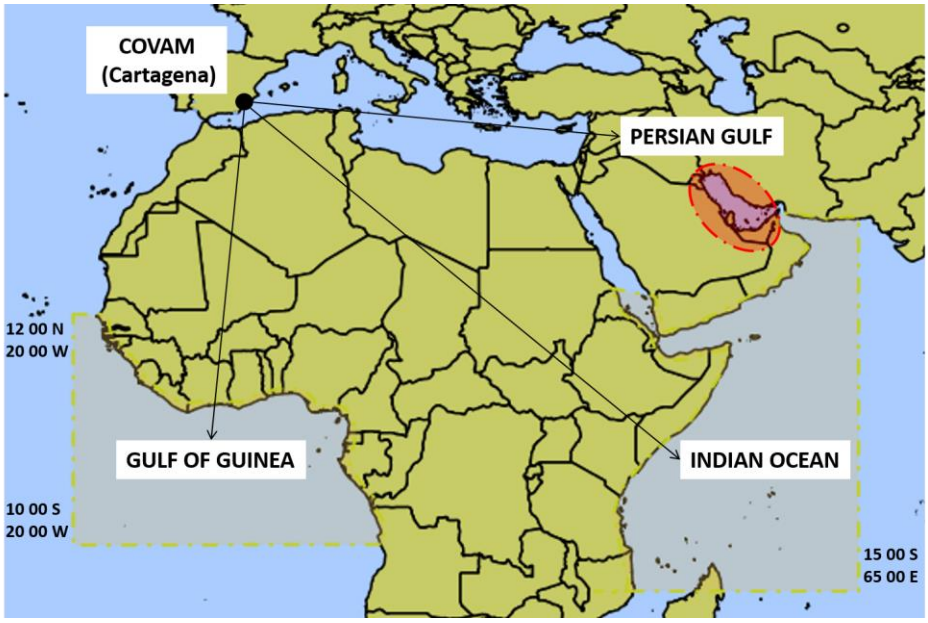


Figura 2-23. Zonas de interés nacional (fuente: [51])

En definitiva, las misiones que se le asignan al COVAM son las siguientes:

- Se encarga de monitorear las misiones asignadas por el Almirante de Acción Maritima (ALMART), por el Comandante del Mando Operativo Marítimo (CMOM) y las ejecutadas por unidades de la FAM. También es responsable de dirigir operativamente estas misiones cuando sea necesario, y tomar medidas inmediatas ante cualquier incidente o situación operativa que requiera acción.
- Busca estar al tanto de la situación táctica del entorno marítimo a través de la construcción de su sistema de conocimiento de la situación (*Maritime Situational Awareness* o MSA) mediante la utilización de diferentes sistemas de control de tráfico. En definitiva, busca mantener una robusta RMP (*Recognized Maritime Picture)*.
- Establece relaciones con otras autoridades, unidades de la Armada y equivalentes en otras armadas de países aliados, así como con organizaciones y autoridades civiles. Se convierte en un punto de contacto entre la Armada y la comunidad marítima.

Para realizar sus diferentes cometidos el COVAM cuenta con una herramienta fundamental para el desarrollo de su trabajo que es el sistema de identificación automático (AIS). Este sistema da información de diferentes orígenes: barcos, aeronaves en la mar, señales satelitales y estaciones fijas en tierra. Esta información permite conocer aspectos cruciales sobre los barcos, como su identidad, origen y destino, dirección y velocidad. Además, también se obtiene información por otros medios y sensores, como unidades militares nacionales y extranjeras, y barcos de pesca que, aunque no emiten señales AIS, pueden ser localizados a través del sistema de balizamiento que están obligados a utilizar. Toda esta información es enriquecida con datos adicionales, tales como bases de datos relacionadas con el mar, historiales de barcos, empresas marítimas y de pesca [50].

2.5 Subdivisión del espacio geográfico

La subdivisión del espacio geográfico para conseguir la localización de una zona determinada es un hecho en la actualidad. Abarca desde las proyecciones cartográficas, que consisten en representaciones de la Tierra en planos a través de una serie de reglas matemáticas (proyección de Mercator, 1569), hasta la subdivisión jerárquica que consiste en dividir el espacio geográfico en una serie de regiones de diferentes tamaños y niveles de detalle. Una de las técnicas de subdivisión jerárquica más antiguas y conocidas es la técnica de subdivisión de *Thiessen*, desarrollada por *Alphonse Thiessen* en 1911.

Entre las celdas más comunes podemos encontrar:

- Celdas cuadradas (S2): Una de las técnicas más comunes de subdivisión geográfica. Cada celda cuadrada se identifica mediante su posición en un sistema de coordenadas cartesianas. Esta técnica es fácil de implementar y permite una representación precisa de los datos, pero sufre de distorsión en áreas cercanas a los polos o en zonas de gran extensión.
- Celdas hexagonales: se identifica mediante un código único que refleja su posición en el espacio geográfico. Esta técnica es eficiente para manejar datos geográficos a nivel mundial con precisión y eficiencia, y es utilizada en herramientas como las celdas H3 de Uber [5]. De esta técnica hablaremos posteriormente ya que será el tipo de celda que usaremos en la realización de este TFG.
- Celdas poligonales: se pueden utilizar polígonos como celdas de subdivisión geográfica, como el caso de la técnica de *Voronoi*. En la Figura 2-24 se puede ver que divide el espacio geográfico en una serie de polígonos que contienen todos los puntos que están más cerca de un punto dado que de cualquier otro punto [52].

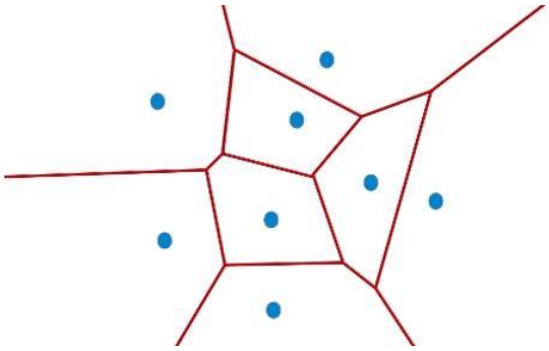


Figura 2-24. Diagrama de Vornoi (fuente: [52])

- Celdas basadas en subregiones: se pueden utilizar subregiones existentes, como estados, provincias o condados, como celdas de subdivisión geográfica.

2.5.1 Celdas hexagonales: H3 Uber

Las celdas H3 son una herramienta de análisis geoespacial desarrollada por Uber para representar y analizar datos geográficos de manera precisa y eficiente. Esta herramienta se basa en el uso de hexágonos regulares que dividen el espacio geográfico en áreas uniformes y analizan la distribución de puntos de interés, como paradas de transporte, restaurantes, y otros lugares relevantes [53].

La idea detrás de las celdas H3 es utilizar una proyección de icosaedro modificada para dividir el espacio geográfico en celdas regulares de tamaño variable. Los hexágonos tienen una única distancia entre su centro y el de los adyacentes a diferencia de las celdas cuadradas, con dos distancias, o triangulares que tienen tres como podemos observar en la Figura 2-25.

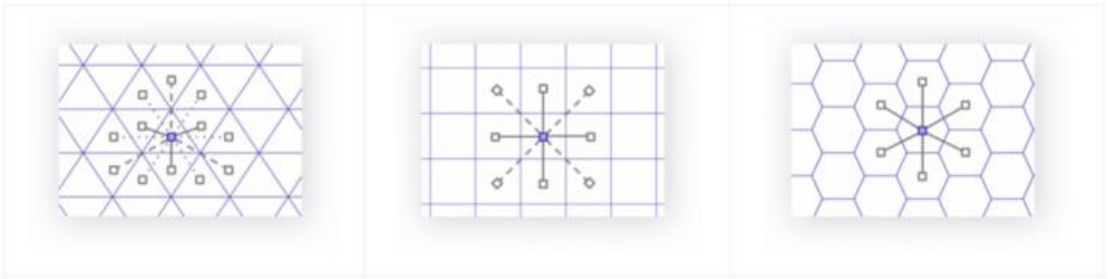


Figura 2-25. Distancias entre celdas (fuente: [53])

Cada celda está identificada por un código hexadecimal único que refleja su posición en el espacio geográfico. Esto permite a los usuarios ver y analizar la distribución de los datos en una visualización fácil de entender, lo que facilita la toma de decisiones basadas en datos.

Una de las principales ventajas de las celdas H3 es su capacidad para manejar datos geográficos a nivel mundial con precisión y eficiencia. A diferencia de otras herramientas de análisis geoespacial, las celdas H3 no sufren de distorsión en áreas cercanas a los polos o en zonas de gran extensión. Además, el uso de celdas regulares permite una representación precisa de los datos, incluso en áreas densamente pobladas. Otra ventaja de las celdas H3 es su capacidad para manejar datos a diferentes niveles de resolución [54]. La herramienta permite dividir el espacio geográfico en celdas de diferentes tamaños, desde celdas grandes que abarcan grandes áreas, hasta celdas pequeñas que representan áreas muy precisas. Esto permite a los usuarios analizar los datos a diferentes niveles de detalle (Figura 2-26), lo que es especialmente útil para proyectos que requieren una gran precisión geográfica.

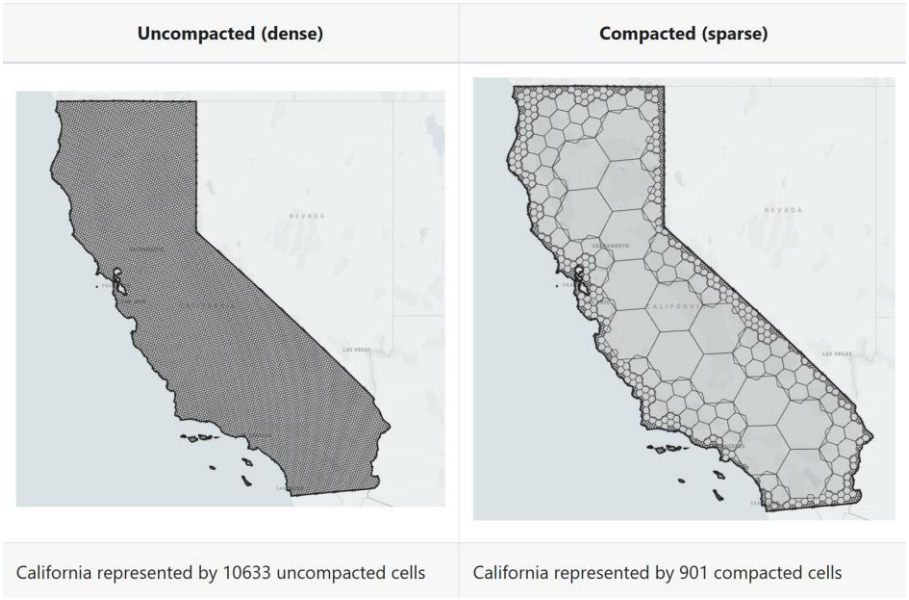


Figura 2-26. California subdividida en celdas H3 de diferentes tamaños (fuente: [55])

H3 Resolution	Average Hexagon Area (km^2)	Average Hexagon Edge Length (km)	Number of unique indexes
0	4,250,546.8477000	1,107.712591000	122
1	607,220.9782429	418.676005500	842
2	86,745.8540347	158.244655800	5,882
3	12,392.2648621	59.810857940	41,162
4	1,770.3235517	22.606379400	288,122
5	252.9033645	8.544408276	2,016,842
6	36.1290521	3.229482772	14,117,882
7	5.1612932	1.220629759	98,825,162
8	0.7373276	0.461354684	691,776,122
9	0.1053325	0.174375668	4,842,432,842
10	0.0150475	0.065907807	33,897,029,882
11	0.0021496	0.024910561	237,279,209,162
12	0.0003071	0.009415526	1,660,954,464,122
13	0.0000439	0.003559893	11,626,681,248,842
14	0.0000063	0.001348575	81,386,768,741,882
15	0.0000009	0.000509713	569,707,381,193,162

Tabla 2-2. Resolución celdas H3 (fuente: [54])

En función de la necesidad, el número de celdas puede llegar a ser mucho mayor y de un tamaño mucho más reducido, si lo que se busca es precisión. Por el contrario, los hexágonos pueden ser más grandes y con un menor número de celdas. Las celdas de mayor tamaño son las de nivel 0 que van desde la celda 0 situada en el extremo norte hasta la 121 situada más al sur. En cada subnivel (Tabla 2-1) el hexágono de nivel superior pasa a subdividirse en hexágonos con un área de tamaño *1/7* del anterior, y el lado de dichos hexágonos será de $1/\sqrt{7}$, llegando hasta el nivel 15 donde el área de las celdas es de $1\ m^2$ y su lado de 5 *dm* [5].

Figura 2-27. Subregiones vs H3 (fuente: [56])

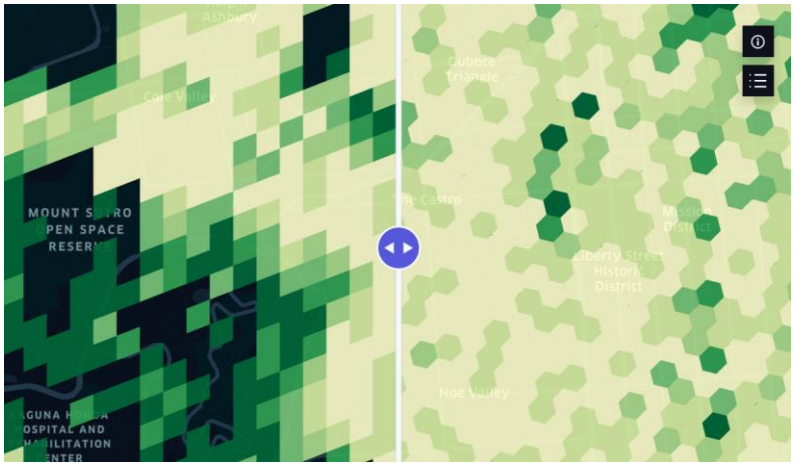


Figura 2-28. S2 vs H3 (fuente: [57])

En la Figura 2-27 y en la Figura 2-28 se pueden ver las diferencias que habría entre celdas en función de subregiones (código ZIP) las cuales son mucho menos precisas debido a su gran tamaño, y las celdas cuadradas (S2) donde ocurre algo similar, pero a menor escala. Es por ello por lo que una de las principales ventajas que presentan este tipo de celdas es la distancia al centro de sus vecinos (ya mencionado anteriormente), donde en las celdas H3 siempre va a ser menor y de valor único mientras que en el resto irá cambiando en función de la geometría de estas.

Las celdas H3 han supuesto una mejora en la localización de objetos en el espacio, además de que el flujo de datos es mucho más sencillo y ágil que usando otro tipo de celdas. Es por ello por lo que para el desarrollo de este TFG usaremos las celdas H3 como principal sistema de localización de buques.

2.6 Trabajos previos

A continuación, se expondrán diferentes estudios realizados en el ámbito de este TFG. Principalmente están basados en predicción de tipo de buque en base a diferentes técnicas de inteligencia artificial y en el análisis de las zonas geográficas marítimas de interés.

Los dos primeros estudios (apartados: 2.6.1 y 2.6.2) se engloban dentro del proyecto que realizó el Centro Universitario de la Defensa en la Escuela Naval Militar (CUD-ENM) junto con el COVAM. La ausencia del campo *shiptype* en los mensajes AIS es una realidad y son muchos los estudios relacionados con este tema. El fin principal es, mediante métodos de IA, conseguir una predicción que sea fiable del

tipo de buque con el objetivo de mantener una RMP robusta; la institución encargada de esto en España, como ya se mencionó en el apartado 2.4.1, es el COVAM.

Si bien es cierto que todos los estudios relacionados en este ámbito se centran en el análisis de parámetros estáticos con resultados muy favorables, se pueden encontrar *papers* en los que se trabajan con parámetros dinámicos, pero no hay ninguno que se centre en el análisis de las celdas H3 como fuente principal para predecir el tipo de buque.

2.6.1 Predicción de tipo de buque utilizando datos AIS y técnicas de inteligencia artificial

Durante el curso 2021-2022, el A.N. Gonzalo Rodríguez Casajús realizó este proyecto como trabajo de fin de grado [4]. El objetivo era aplicar técnicas de inteligencia artificial para mejorar el conocimiento del entorno marítimo y ayudar en la detección de anomalías en el comportamiento de los barcos.

Se propuso la aplicación de técnicas de aprendizaje automático supervisado para predecir el valor tipo de barco utilizando información AIS. En particular se utilizó el algoritmo de *Random Forest*, llegando a la conclusión que los datos AIS más efectivos para la predicción del tipo de buque eran los datos estáticos de los propios barcos.

2.6.2 Análisis de los sistemas de indexado geoespacial para el Conocimiento del Entorno Marítimo

Durante el curso 2020-2021, el A.N. Víctor Alonso Aller realizó este trabajo de fin de grado [58]. En él se llevó a cabo un estudio sobre los métodos de indexación geoespacial y las características del *big data* en el entorno marítimo. Se evaluaron varios sistemas y se eligieron las celdas H3 de Uber por sus características. Utilizando el sistema gestor de datos MySQL y la aplicación de este sistema de indexación, se analizaron datos reales realizando consultas espaciales, llegando a la conclusión de que la utilización de este sistema de indexación es mucho más rápida.

2.6.3 Método de clasificación de tipos de buque basado en información AIS y SAR.

Este trabajo [59] se centra en el desarrollo de un método de clasificación de tipos de barcos basado en la información de los sistemas de identificación automática (AIS) y radares de apertura sintética, SAR (*Synthetic Aperture Radar*). La metodología utilizada incluye la extracción de características de la información de AIS y SAR, seguida de un análisis de los datos utilizando técnicas de aprendizaje automático: regresión logística y el algoritmo de vectores de soporte. El objetivo principal de esta investigación es mejorar la precisión y la eficiencia en la identificación de los tipos de barco. Sus autores destacan la importancia de una clasificación precisa y eficiente debido a la creciente necesidad de conocer la información detallada sobre los barcos y su carga para fines de seguridad marítima, planificación de rutas y gestión de la flota, en definitiva, conocimiento del entorno marítimo (CEM).

2.6.4 Predicciones de tráfico marítimo usando técnicas de Machine Learning y datos AIS

Los autores presentan en [60] una metodología innovadora para predecir la posición futura de los barcos utilizando algoritmos de aprendizaje automático y datos AIS. La metodología combina técnicas de aprendizaje supervisado y no supervisado para predecir la posición futura de los barcos. Además, los autores también evalúan la precisión de la metodología propuesta y demuestran que es efectiva para predecir la posición futura de los barcos en tiempo real.

El estudio también discute cómo la predicción del tráfico marítimo puede ser utilizada en una variedad de aplicaciones prácticas, como el monitoreo de la seguridad marítima, la gestión de la navegación o la planificación de la logística marítima. En general, el estudio proporciona una solución innovadora y prometedora para predecir el tráfico marítimo utilizando algoritmos de aprendizaje automático y datos AIS.

3 DESARROLLO DEL TFG

En este capítulo, primero se presentará el entorno de trabajo que se ha utilizado, así como el lenguaje de programación. Estos formarán las herramientas a partir de las cuales se realizará el análisis de los datos AIS, de los que también se explicará su origen y las diferentes medidas que se han tomado para conseguir aumentar la eficiencia de estos, con el fin de conseguir los datos más puros posibles. Adicionalmente se explicará el uso de las celdas H3 como posible parámetro para mejorar la predicción de tipo de buque y la metodología que se ha utilizado para el desarrollo del modelo.

Pará finalizar se expondrá un resumen de los todos experimentos realizados en base a las diferentes fuentes de información: datos estáticos, datos cinemáticos y celdas H3.

3.1 Entorno de trabajo y software empleado

3.1.1 Jupyter Lab

El entorno de trabajo que se ha utilizado para el desarrollo del TFG es *Jupyter Lab* [61]. Se trata de una interfaz que permite configurar y organizar el flujo de datos de una manera sencilla. *Jupyter Lab* es el sucesor oficial de *Jupyter Notebook* creado por la organización sin ánimo de lucro *Proyecto Jupyter* en el año 2015 [62]. La principal diferencia con su predecesor es la simplicidad que se ha logrado con *Jupyter Lab*: editores de texto más sencillos y mayor facilidad para acceder a carpetas externas del programa como puede ser *Google Drive.*

Entre los principales beneficios del programa están:

- Código abierto: el código fuente se encuentra disponible para todos los usuarios, lo que permite modificarlo y compartirlo, formando una red donde los usuarios se van retroalimentando con las diferentes mejoras.
- *Software* gratuito.
- Soporta más de cincuenta lenguajes de programación.

3.1.2 Lenguaje de programación: Python

Python [63] es hoy en día el lenguaje de programación con mayor proyección. Gracias a su comunidad activa y a ser de código abierto, su empleo crece cada día. Su estilo flexible permite al usuario una comprensión de la sintaxis más accesible y mayor facilidad a la hora de declarar atributos. Por otro lado, es un lenguaje multiplataforma que permite utilizarlo con diferentes sistemas operativos como pueden ser *Linux, Mac OS, Windows*.

Es por ello por lo que se ha elegido este lenguaje para el desarrollo del TFG. Adicionalmente, como ya se mencionó en el apartado 3.1.1, *Jupyter Lab,* a pesar de ser capaz de trabajar con más de cincuenta lenguajes, se creó para ser utilizado con *Python.*

3.1.3 Base de datos: SQLite3

Se hace necesario disponer de una base de datos que almacene de forma persistente toda la información que se genera es necesario. En el desarrollo del TFG se está trabajando con cientos de millones de mensajes AIS, es por ello por lo que se necesita de un sistema de almacenamiento de información. Para ello se ha elegido el sistema de base de datos *SQLite3* [64]. En dicho sistema, la información se organiza en tablas y el tiempo de consulta a los datos es mucho menor que sí se realizasen consultas directamente al archivo *CSV* [65].

3.1.4 Librerías

La Tabla 3-1 muestra las diferentes librerías que se han utilizado.

Librería	**Funciones**
Pandas	Análisis y procesamiento de datos.
SQLite3	Consulta SQL (*Structured Query Language*).
NumPy	Cálculo numérico gran número de datos.
Sklearn	Aprendizaje automático (*Random Forest, kNN*).
SeaBorn	Visualización de datos.
Matplotlib	Visualización de datos.
TQDM	Barras de progreso.
Imblearn	Aprendizaje automático (equilibrio datos).
H3	Funciones relacionadas con celdas H3.

Tabla 3-1. Librerías empleadas

3.2 Datos AIS

Como ya se adelantó en el apartado 2.4.1, los datos AIS son la principal fuente de información utilizada por el COVAM para el análisis de la RMP. Los equipos que generan estos datos fueron creados en la década de los 90 y fueron regulados inicialmente por la Organización Marítima Internacional (OMI). Posteriormente en el año 2002 con la firma del convenio SOLAS (*Safety of Life at Sea*) se comenzó a obligar a portar transpondedores AIS a los diferentes buques en función de sus características:

- Transpondedor clase A: este equipo es solo obligatorio en barcos de más de 300 toneladas en tránsitos intercontinentales. Ofrece una mayor capacidad de transmisión de datos y mayor alcance.
- Transpondedor clase B: este equipo fue diseñado fundamentalmente para embarcaciones de recreo. Se trata de un equipo más sencillo y económico que el de clase A. Al contrario que los transpondedores de clase A, estos no son obligatorios.

La información que nos ofrece el sistema AIS puede ser de dos tipos: estática o dinámica. La primera hace referencia a los datos del buque que no varían a lo largo del tiempo como la eslora, manga o bandera del país y se transmite cada seis minutos aproximadamente, mientras que los datos dinámicos varían en función de la situación. Estos hacen referencia a la velocidad o al rumbo y se transmiten entre cada dos y diez segundos [66].

Se pueden encontrar más de 27 tipos de mensajes AIS [67], aunque para el desarrollo de este TFG se han utilizado los siguientes cuatro tipos de mensajes:

- Mensajes de tipo 1, 2 y 3: estos mensajes contienen la información dinámica del buque, como la velocidad, rumbo o posición geográfica.
- Mensajes de tipo 5: ofrecen la información estática del buque como el calado, eslora o manga. Estos dos últimos valores vienen desglosados en las medidas *A, B, C* y *D* como se puede ver en la Figura 3-1. En la imagen se muestra las diferentes medidas y con un círculo la ubicación del transpondedor AIS.
-

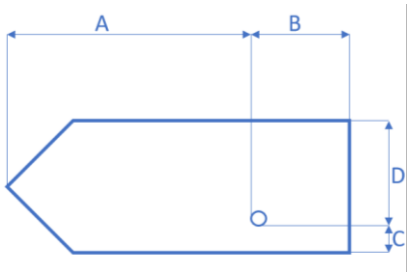


Figura 3-1. Referencias a las dimensiones del buque de los mensajes AIS (fuente: propia)

En este TFG se han utilizado archivos CSVs que contienen la información de los diferentes mensajes. Para crear los distintos CSVs (tanto los estáticos como los dinámicos) una plataforma web de monitorización del CUD-ENM se encarga de procesar todos los mensajes AIS recibidos del COVAM. El proceso comienza con el envío de los mensajes AIS por parte del COVAM, y la plataforma web del CUD-ENM analiza esos datos AIS, los depura, los transforma en otras estructuras y los enriquece con otras fuentes de datos (bases de datos registrales, zonas de fondeo, celdas H3, etc). Más en concreto, si hay campos sin cubrir como el IMO o el MMSI, la base de datos registral puede servir para completar el campo restante. Además, se añade un campo nuevo, la celda H3 de nivel nueve, calculada a partir de la información de latitud y longitud. Una vez finalizado este proceso se generan archivos con datos estáticos y otros con datos dinámicos.

3.2.1 Atributos estáticos

Los archivos CSV con datos estáticos contienen la información de los mensajes AIS tipo 5 con los siguientes campos*:*

- Campo A: distancia desde el transpondedor AIS y la proa en metros (*to_bow*).
- Campo B: distancia desde el transpondedor AIS hasta la popa en metros (*to_stern*).
- Campo C: distancia desde el transpondedor AIS hasta la banda de babor en metros (*to_port).*
- Campo D: distancia desde el transpondedor AIS hasta la banda de estribor en metros (*to_starboard).*
- *Draught*: calado del buque en metros.
- *Shiptype*: tipo de buque escrito explícitamente, no usando el código numérico.

Como se ha mencionado anteriormente los atributos estáticos son aquellos que hacen referencia a parámetros invariables a lo largo del tiempo. Se ha creado la función *statics* a partir de la cual se definen las variables de la Tabla 3-2, utilizando los valores mencionados anteriormente.

Variable	Descripción
$len = A + B$	Eslora
$wid = C + D$	Manga
$draught$	calado
$ldivw = \frac{len}{wid}$	Cociente eslora y manga
$ldivd = \frac{len}{draught}$	Cociente eslora y calado
$wdivd = \frac{wid}{draught}$	Cociente manga y calado
$area = len * wid$	Área de la cubierta
$grith = len + wid$	Suma eslora y manga
$aml = len * draught$	Área longitudinal sumergida
$amt = wid * draught$	Área transversal sumergida
$vs = len * wid * draught$	Volumen sumergido
$aol = \frac{a}{len}$	Proporción de *a* sobre la eslora total

Tabla 3-2. Variables estáticas

El código de la función *statics* es el siguiente:

```
def statics(df):
    df['len']=df.a+df.b
    df['wid']=df.c+df.d
    df['ldivw']=df.len/df.wid
    df['ldivd']=df.len/df.draught
    df['wdivd']=df.wid/df.draught
    df['area']= df.len*df.wid
    df['grith']= df.len+df.wid
    df['aml']=df.len*df.draught
    df['amt']=df.wid*df.draught
    df['vs']=df.len*df.draught*df.wid
    df['aol']=df.a/df.len
    return df
```

3.2.2 Atributos dinámicos

En cuanto a los atributos dinámicos se han utilizado los archivos CSV con los datos dinámicos que emplean como fuente de información los mensajes AIS de tipos 1, 2 y 3. En particular, dichos archivos cuentan con la siguiente información:

- *TIMESTAMP*: fecha y hora del mensaje.
- *SOG*: *speed over ground*, velocidad sobre el fondo.
- *COG*: *course over ground*, rumbo sobre el fondo.
- *Latitude*: latitud geográfica, utilizado para cálculos cinemáticos.
- *Longitude*: longitud geográfica, utilizado para cálculos cinemáticos.
- *Shiptype*: código numérico que define el tipo de buque (Tabla IV-1).
- *H3_9*: celdas H3 resolución nivel nueve (sistema numérico hexadecimal) campo incorporado por el CUD-ENM.

Los atributos dinámicos que se han utilizado en este TFG se pueden separar en dos grupos: por un lado, están los calculados a partir de los datos cinemáticos del barco, para lo que se utiliza la velocidad, el rumbo, la fecha y hora del mensaje y la posición geográfica. Por otro lado, se han calculado otros atributos en relación con la localización del buque en base a las celdas H3.

3.2.3 Cinemáticos

Como se ha expuesto anteriormente se han utilizado los atributos relacionados con la cinemática del buque para el cálculo de diferentes variables que posteriormente servirán para calcular los estadísticos cinemáticos de cada barco. Para ello se ha creado la función *cinematics* con la que se calculan las variables de la Tabla 3-3, estos valores son únicos para cada situación del barco (cada mensaje recibido).

El código utilizado para la función *cinematics* es el siguiente:

```
def cinematics(df):
    #tiempo
    df = df.sort_values(by='tim')
    df.tim = pd.to_datetime(df.tim,unit='ms')
    df['dtime'] = df.tim.diff().dt.seconds
    hours = df.dtime/3600
   #lineal
    dy = np.radians(df.lat.diff())*6373/1.852
    dx = np.radians(df.lon.diff())*6373/1.852
    df['nmi'] = np.sqrt(dx**2 + dy**2)
    df['sog_hat'] = df.nmi/hours
    df['sog_err'] = (df.sog_hat-df.sog)/(df.sog+0.1)*100
    df['acc_hat'] = df.sog_hat.diff()/hours
    df['jerk_hat'] = df.acc_hat.diff()/hours
    df['acc'] = df.sog.diff()/hours
    df['jerk'] = df.acc.diff()/hours
    #angular
    deg_norm = df.cog.diff()%360
    deg_diff = pd.concat([360-deg_norm, deg_norm], axis=1).min(axis=1)
    df['omega'] = np.radians(deg_diff)/hours
    df['alpha'] = df.omega.diff()/hours
    df['zeta'] = df.alpha.diff()/hours
    return df
```

Variables	Descripción
dtime	Distancia temporal entre mensajes
nmi	Distancia espacial entre mensajes
sog_hat	Velocidad a partir de la distancia espacial entre mensajes
sog_err	Error respecto a la velocidad AIS
acc	Aceleración a partir de la distancia temporal
acc_hat	Aceleración a partir de la distancia espacial entre mensajes
jerk	Sobreaceleración a partir de la distancia temporal
jerk_hat	Sobreaceleración a partir de la distancia espacial entre mensajes
omega	Velocidad caída de rumbo
alpha	Aceleración caída de rumbo
zeta	Sobreaceleración caída de rumbo

Tabla 3-3. Atributos cinemáticos

Una vez calculados, se aplica la función *aggregate*, mediante la cual se determinan los parámetros estadísticos de cada una de las variables de la Tabla 3-3. Dichos parámetros son los siguientes:

- ***avg***: la media de los datos.
- ***sdv***: desviación estándar.
- ***iod*:** cociente de la varianza entre la media.
- ***q50***: percentil 50 (mediana).
- ***skw***: cálculo asimetría conjunto de datos.
- ***krt***: curtosis de los datos.

Un ejemplo de la función *aggregate* es el que aparece a continuación, los parámetros calculados son los referentes a la variable aceleración (*acc*):

```
def aggregate(df):
    #aceleración
    df['avg_acc'] = df['acc'].mean()
    df['sdv_acc'] = df['acc'].std()
    df['iod_acc'] = df['acc'].var()/df['avg_acc']
    df['q50_acc'] = df['acc'].quantile(0.50)
    df['skw_acc'] = df['acc'].skew()
    df['krt_acc'] = df['acc'].kurt()
    return df
```

3.2.4 Celdas H3

Como se mencionaba anteriormente, el objetivo principal de este TFG consiste en analizar la influencia de nuevos parámetros estadísticos basados en la posición del buque con la mejora de la predicción del tipo de barco. Las celdas H3 utilizadas son de nivel nueve, cuya resolución corresponde a $0{,}10533\ km^2$ como se mencionó en el apartado 2.5.1 de este trabajo.

Para el cálculo de los atributos se han utilizado celdas H3 de nivel nueve y de nivel siete, cuya resolución corresponde a $5{,}16\ km^2$. El fin por el que se ha decidido usar celdas de diferentes tamaños es para diferenciar patrones de buques, por ejemplo: barcos que trabajan en un área más pequeña o barcos que hacen recorridos entre puertos más lejanos.

Para el cálculo de las celdas H3 de resolución siete, se ha utilizado la función *h3_to_parent* definida en la librería H3. Con ella se determina la celda de nivel inferior (menor resolución) sobre la que se está aplicando. En este TFG se aplicará dos veces para llegar a las celdas deseadas. La Figura 3-2 muestra un ejemplo de celdas de diferente tamaño en una misma zona geográfica, en este caso se representan celdas de nivel 5 y 6.

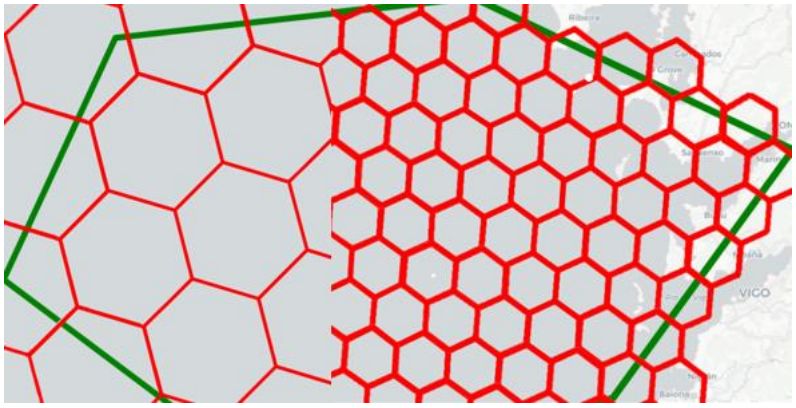


Figura 3-2. Ejemplo de celdas de diferente tamaño en la costa gallega (fuente: propia)

Para ilustrar lo anterior se pueden analizar las diferencias entre patrones de navegación de un remolcador y un mercante. Ambas clases de barcos tienen altos periodos de movimiento, sin fondeos, lo que se traduce en que la corredera de los dos tipos de barco a lo largo de un día puede marcar un alto número de millas navegadas. La diferencia entre ambos barcos estará en su trayectoria. El mercante llevará una trayectoria rectilínea lo que hará que recorra un mayor número de celdas H3 diferentes, mientras que el remolcador navegará en una misma área que quedará limitada a dos o tres celdas.

Las variables calculadas para cada barco han sido las siguientes:

- ***h3_7***: celdas H3 de resolución siete.
- ***freq9***: frecuencia de repetición de las celdas de resolución nueve.
- ***freq7***: frecuencia de repetición de las celdas de resolución siete.

A partir de estas variables se han calculado los atributos de la Tabla 3-4 que hacen mención solamente a los parámetros de las celdas de nivel 9, se calcularían todos ellos también para celdas de resolución 7 (*h3r7*).

ATRIBUTOS	DESCRIPCIÓN
h3_meanfreq	Media frecuencia de repetición.
h3_maxfreq	Frecuencia máxima de repetición.
h3_suma	Sumatorio de las celdas diferentes por las que navega.
h3_skwfreq	Asimetría del conjunto de datos.
h3_krtfreq	Curtosis de los datos.
h3_q50freq	Percentil 50.
h3_stdfreq	Desviación estándar.
h3_iod	Cociente de la varianza entre la media.
h3_maxdistance	Distancia máxima entre celdas de nivel 9.
h3_distanceFL	Distancia entre primera y última celda de nivel 9.
h3_difdistance	Diferencia entre *h3_maxdistance* y *h3_distanceFL*.
h3_pathdistance	Trayectoria del buque en base a las celdas H3.

Tabla 3-4. Atributos dinámicos respecto a las celdas de nivel 9

Para su utilización en el algoritmo se ha creado la función *statistics* mediante la cual se calculan las frecuencias máximas y medias de cada buque respecto a las celdas de nivel nueve y nivel siete por las que navega cada barco. A continuación, se muestra la función para el cálculo de dichos atributos para las celdas de resolución nueve.

```
def statistics9(df):
    df['h3_meanfreq'] = df['freq'].mean()
    df['h3_stdfreq'] = df['freq'].std()
    df['h3_iod'] = df['freq'].var()/ df['h3_meanfreq']
    df['h3_q50freq'] = df['freq'].quantile(0.50)
    df['h3_skwfreq'] = df['freq'].skew()
    df['h3_krtfreq'] = df['freq'].kurt()
    return df
```

Además, para el cálculo de las distancias entre celdas se ha creado la función *max_distance* a partir de la cual se calcula la máxima distancia que hay entre las dos celdas más alejadas por las que ha pasado un barco. Para el cálculo de la distancia entre celdas se usa la función predeterminada *h3_distance* de la librería H3. Como se puede observar en las siguientes líneas de código, mediante la función mencionada anteriormente se calcula la distancia entre dos celdas, dicha función queda limitada a distancias relativamente cercanas. Es por ello, por lo que cuando la función no es capaz de calcular dicha distancia se asigna un valor de -1 al valor de la distancia máxima como resultado de que hay distancias muy lejanas. Que las distancias sean grandes resulta incoherente cuando se habla de distancias navegadas por barcos de superficie en periodos como máximo de un día.

```
def max_distance(celdas, h3distance):
    for i in range(0, len(celdas)-1):
        for j in range(i+1, len(celdas)):
            try:
                dist = h3.h3_distance(celdas[i], celdas[j])
                if dist > h3distance:
                    h3distance = dist
            except:
                h3distance = -1
                return h3distance

    return h3distance
```

Antes de emplear estos nuevos parámetros en el modelo, se ha hecho un análisis de cómo se comportan estos parámetros en función del tipo de barco.

A continuación, se expondrán diferentes casos donde se observa el comportamiento de los barcos en función de su clase.

1. Distancia máxima entre celdas

La Figura 3-3 muestra la distancia máxima entre celdas H3 de resolución 7 en función del tipo de buque. En el gráfico se puede observar cómo los comportamientos de los barcos en función de su clase generan diferentes patrones que son coherentes con los comportamientos de dichos barcos en la vida real. En una visión general se puede observar que los barcos con mayores distancias son los mercantes y petroleros, sus funciones como buques de transporte hacen que tengan que recorrer grandes distancias, además el margen de distancias es mucho mayor que el de otras clases. En el caso de los mercantes desde 100 hasta 210 aproximadamente. Esto es causado por el gran número de barcos y trayectorias diferentes que pueden ir de tránsitos más cortos (Cádiz-Canarias) a recorridos trasatlánticos (Lisboa-Nueva York).

En el otro extremo se encuentran los remolcadores, con distancias mucho menores, de nuevo coherente con su función como buque de remolque para ayudar a entrar en puerto al resto de barcos. Si bien se puede observar en la parte derecha de la Figura 3-3 que la dispersión que hay en distancia máxima en esta clase es mucho menor respecto a los mercantes/petroleros, se pueden ver valores fuera de esos márgenes, y esto puede ser debido a dos motivos: barcos de remolque en alta mar (muy minoritarios) o datos anómalos (*outliers*) los cuales se tratarán más adelante.

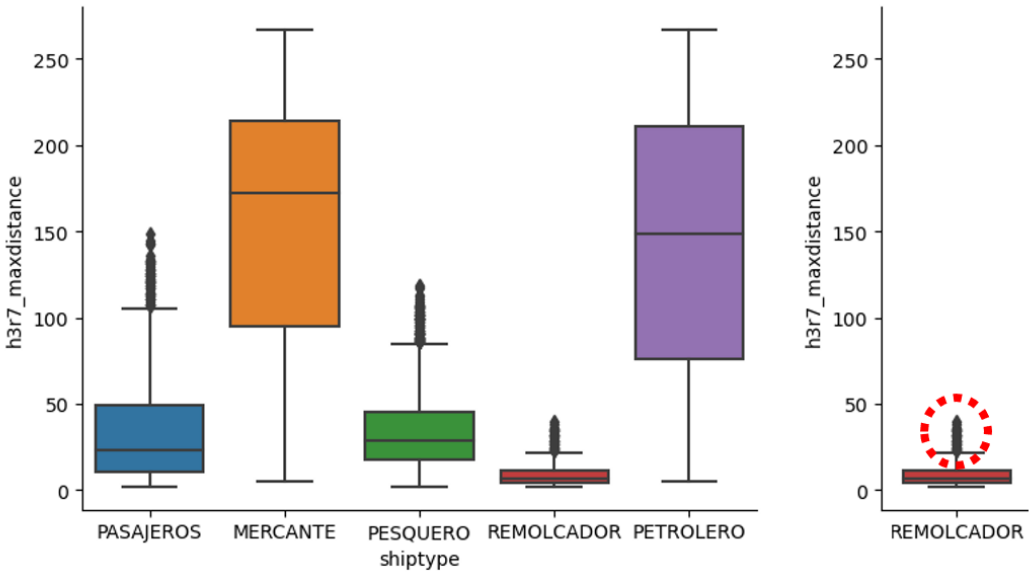


Figura 3-3. Representación gráfica de la distancia máxima entre celdas de resolución 7 (fuente: propia)

2. Diferencia de distancias:

La Figura 3-4 muestra tres gráficos. En la esquina superior izquierda se puede ver el mismo gráfico de la Figura 3-3 y en la esquina inferior izquierda está el gráfico que representa la distancia entre las celdas de los primeros y últimos mensajes respectivamente de cada tipo de barco. El fin de utilizar este atributo es analizar si el puerto de destino es diferente al de origen. Y en el lado derecho hay un gráfico que representa la diferencia entre las distancias representadas en el lado izquierdo para cada barco.

Empezando de nuevo con los mercantes y petroleros, se puede ver que esta diferencia en la mayor parte de los casos es de un valor próximo a 0, esto es debido a que los trayectos que hacen no empiezan y terminan en las mismas celdas, de nuevo haciendo referencia al comportamiento de estos barcos es coherente por su labor como barco de transporte que trasladará mercancías de un puerto a otro. Por otro lado, están los remolcadores cuya diferencia es distinta de 0 pero de un valor pequeño debido a que dichos barcos recorren distancias cortas, sus distancias máximas son pequeñas, en función del nivel de resolución pero un mercante puede llegar a recorrer entre 30-40 celdas de nivel 9 mientras que un remolcador navegará entre 5 y 10. Y por último se encuentran los barcos de pasajeros y pesqueros cuyas diferencias son más grandes ya que sus distancias máximas también lo son, pero sus puertos de inicio y final de singladura suelen ser los mismos, por ejemplo ferris en el estrecho de Gibraltar o pesqueros que salen a faenar durante un par de días.

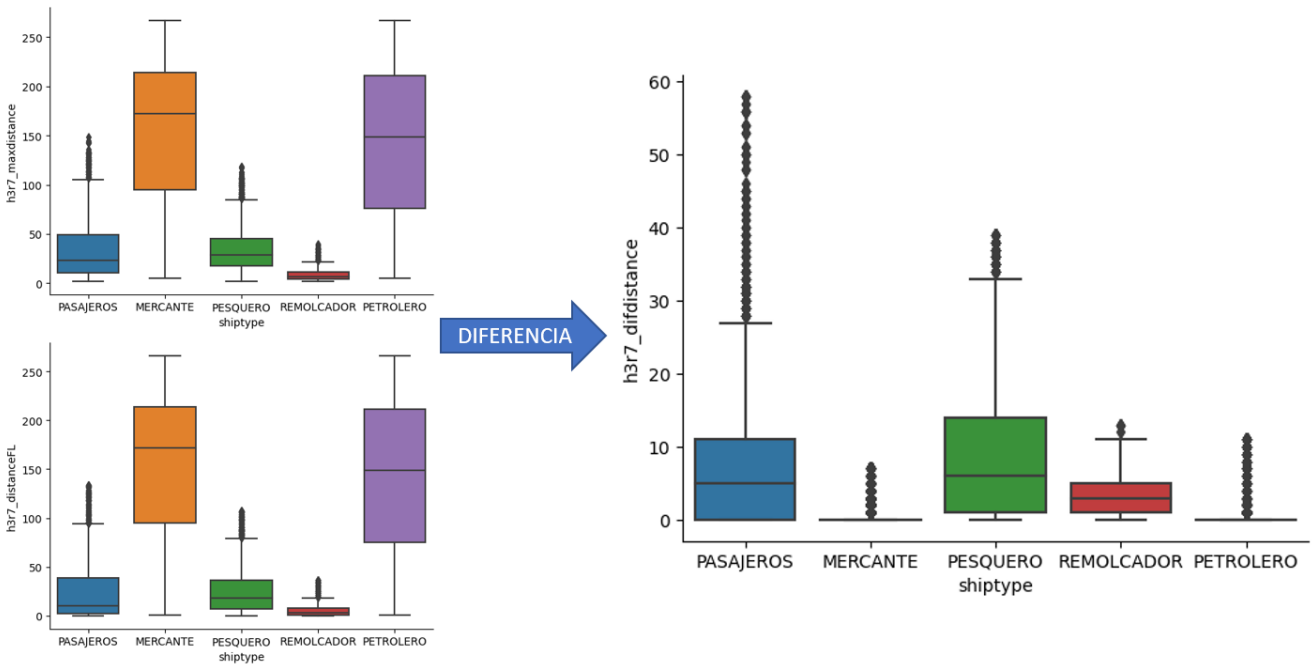


Figura 3-4. Análisis atributos relativos a las distancias entre celdas H3 de resolución 7 (fuente: propia)

3. Trayectorias

El último parámetro por analizar son las trayectorias de dichos barcos en función de las celdas H3 por donde navega. La representación de las trayectorias es en base a las celdas H3 de resolución 7. Como se puede ver en la figura se podrían distinguir tres patrones en base a las distancias recorridas. Los de mayor recorrido que serían los barcos mercantes y petroleros, en segundo lugar, los de media distancia, donde se situarían los pesqueros y los barcos de pasajeros y, por último, los de corta distancia, donde se enmarcarían los remolcadores.

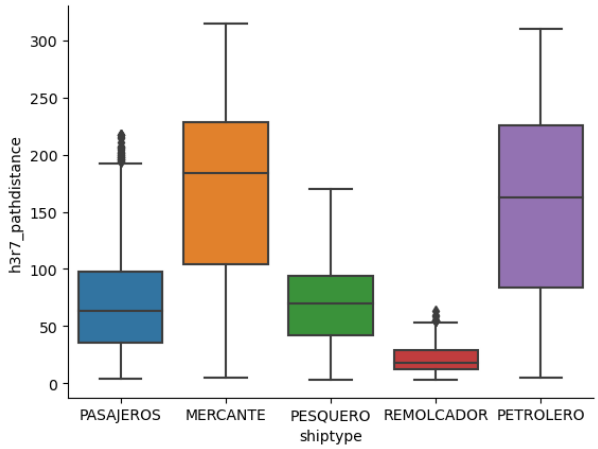


Figura 3-5. Trayectoria en base a celdas H3 de resolución 7 (fuente: propia)

Mediante la adición de estos nuevos parámetros al resto de estadísticos relativos a la frecuencia de aparición de las diferentes celdas se pretende mejorar los resultados obtenidos con el modelo en la predicción de tipo de buque.

3.3 Justificación del algoritmo empleado

En cuanto a los algoritmos empleados en el desarrollo de este TFG se han utilizado dos métodos de aprendizaje supervisado.

3.3.1 Random Forest

Para el análisis de los parámetros estáticos y dinámicos se ha utilizado *Random Forest* (*RF*). Como se puede ver en el apartado 2.3.1.7, es un algoritmo de aprendizaje supervisado basado en la creación de *n* árboles de decisión de manera aleatoria con el fin de evitar problemas de sobreajuste. Se ha partido del estudio realizado en el Trabajo de Fin de Grado mencionado en el apartado 2.6.1. En él se llegaron a las conclusiones de la Tabla 3-5, a partir de las cuales se empezará el análisis de este trabajo.

OPTIMIZACIÓN DE CÓDIGO	
N.º de árboles	100 árboles.
División datos (entrenamiento/test)	75/25.
Oversampling	Solo en los datos de entrenamiento.
Undersampling	No se realizará.
MinMaxScaler	Se utilizará siempre.

Tabla 3-5. Premisas iniciales

Una vez definido el conjunto de datos, se realiza una división entre datos de entrenamiento y de prueba. Para realizar esta división se utilizará la siguiente línea de código, que de forma aleatoria divide el *dataset* en dos grupos; uno formado con el 75% de los datos (*df_train*) y otro con el 25% (*df_test*). Para ello se ha utilizado la función predeterminada *train_test_split* de *Sklearn.*

```
df_train, df_test = train_test_split(df, random_state = 42)
```

Dentro de cada subgrupo se dividirán los datos en variables independientes y dependientes. Los primeros harán referencia a los parámetros estadísticos calculados en las diferentes funciones propias (*aggregates, statistics, max_distance, statics).* Por otro lado, se tendrá la variable dependiente que en este caso será el tipo de buque (*shiptype*). En las siguientes líneas de código se puede ver como se hace esta división, componente *X* hace referencia parámetros dependientes e *y* a independientes.

```
X_train=df_train.drop('shiptype', axis=1)
y_train=df_train.shiptype
X_test=df_test.drop('shiptype', axis=1)
y_test=df_test.shiptype
```

Una vez están definidos ambos grupos de datos y sus respectivos subgrupos *X* e *y*, se aplica *oversampling* solo a los datos de entrenamiento. El objetivo es aumentar el número de muestras de entrenamiento de las clases menos representadas y conseguir una mayor eficiencia por parte del modelo, véase en la Figura 3-6.

Una vez generados los dos subgrupos se procede a la creación del modelo de aprendizaje. Para ello se utiliza la función *Pipeline*, definida en la biblioteca *Sklearn*. Inicialmente mediante la función *MinMaxScaler*, que, como se mencionó en la Tabla 3-5, se utilizará siempre, se escalarán los datos para que su contribución al modelo sea similar y, por otro lado, *RandomForestClassifier* que es el algoritmo de aprendizaje supervisado que se utilizará para la clasificación de los buques, utilizando 100 árboles, también decidido previamente (Tabla 3-5).

```
pipe = Pipeline([('scaler', MinMaxScaler()),('classifier',RandomForestClassifier(n_estimators=100))])
model = pipe.fit(X_train_res, y_train_res)
y_pred = model.predict(X_test)
```

3.3.2 kNN

Por otro lado, se ha utilizado *kNN* para el análisis de las celdas H3 como parámetro independiente. A diferencia de los casos anteriores donde se calculaban los parámetros estadísticos, aquí se ha querido analizar los patrones utilizando todas las celdas por las que pasa un buque. Para ello se ha utilizado el algoritmo de aprendizaje supervisado *kNN*. Su funcionamiento consiste en la predicción utilizando la cercanía entre los datos: se puede profundizar más en la explicación en el apartado 2.3.1.6 de este trabajo. Inicialmente se tuvo que predecir *k* (número de vecinos) óptimo. Una vez definido este *k* se ha creado un modelo siguiendo la misma estructura que con *RF*. A continuación, se presentan las líneas de código que definen el modelo, donde se han utilizado las funciones definidas de *sklearn*, *kNeighborsClassifier* y *MinMaxScaler*.

```
k = 38
pipe = Pipeline([('scaler', MinMaxScaler()),('classifier',KNeighborsClassifier(n_neighbors=k))])
model = KNeighborsClassifier(n_neighbors=k)
model.fit(X_train_res, y_train_res)
y_pred = model.predict(X_test_res)
```

Al igual que en los otros experimentos, se partirán de las premisas siguientes:

- El *k* óptimo será el utilizado.
- Se escalarán los datos del modelo.
- Se aplicará *oversampling* solo a los datos de entrenamiento.
- La división de datos será 75 /25 entrenamiento/test respectivamente.
- No se realizará *undersampling*.

3.4 Optimización del código

3.4.1 Oversampling

Como se mencionó en apartados anteriores inicialmente solo se realizará *oversampling* en los datos de entrenamiento. Durante el análisis de los experimentos se intentará concluir si su aplicación mejora los resultados obtenidos.

Para su realización se utilizará la siguiente línea de código, mediante la función predeterminada de *Python*, *SMOTE* perteneciente a la librería *imblearn*. Con la ejecución de la siguiente sentencia el número de datos aumenta al mismo nivel que el de la clase mayoritaria como se puede ver en la Figura 3-6, consiguiendo un número igual de muestras de todas las clases de buques.

```
oversample=SMOTE(random_state = 42)
X_train_res, y_train_res=oversample.fit_resample(X_train, y_train)
```

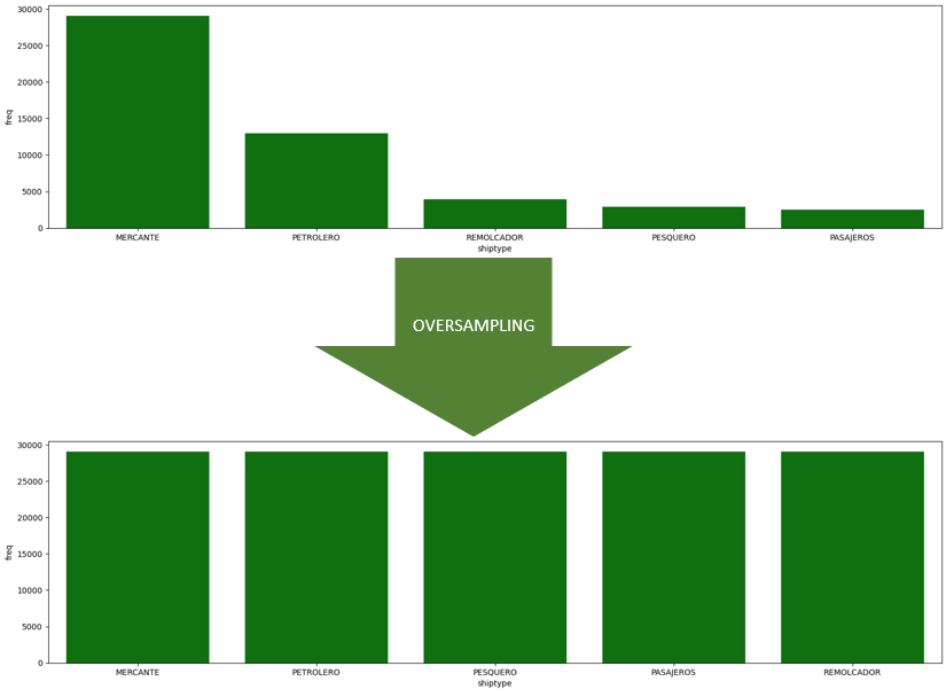


Figura 3-6. Ejemplo de *oversampling* en datos de entrenamiento (fuente: propia)

Por otro lado, se suprime el *undersampling* ya que se ha llegado a la conclusión que eliminar datos reales solo contribuye a empeorar el modelo: realizar solamente *undersampling* limita el número de datos y la información que tiene el modelo para aprender es mucho menor y como consecuencia los resultados también empeoran.

3.4.2 Outliers

El *dataset* con el que se ha estado trabajando contiene un gran número de datos de barcos reales. A veces se da el caso de que un buque puede llegar a tener comportamientos o características que se salen de su patrón. Esto hace que el modelo a la hora de intentar aprender de los comportamientos de dichos barcos asocie patrones que no se corresponden con los de su clase. Por ejemplo: un mercante a una velocidad media de 3 nudos en medio del Océano Índico sería un posible *outlier* por el siguiente motivo: los mercantes son barcos que tienen la finalidad de transportar mercancías de un lugar a otro en el menor tiempo posible, por lo que una velocidad de 3 nudos no es representativa para esta clase de barco, por el contrario, sí que podría ser un pesquero faenando. Es por ello por lo que eliminar estos datos es necesario para evitar que el modelo asociase comportamientos de pesqueros con los de los mercantes en el ejemplo que se planteaba. Adicionalmente filtrando *outliers* se eliminan datos falsos, como velocidades muy altas, imposibles para un barco.

En definitiva, mediante la eliminación de estos *outliers* o datos anómalos se va a conseguir que el modelo afiance los comportamientos de dichos buques de manera más precisa. Por otro lado, la eliminación de dichos parámetros hace que se pierdan un número de muestras a veces muy significativo reduciendo la eficiencia del modelo, es por ello por lo que durante este TFG se intentará analizar sus beneficios e inconvenientes.

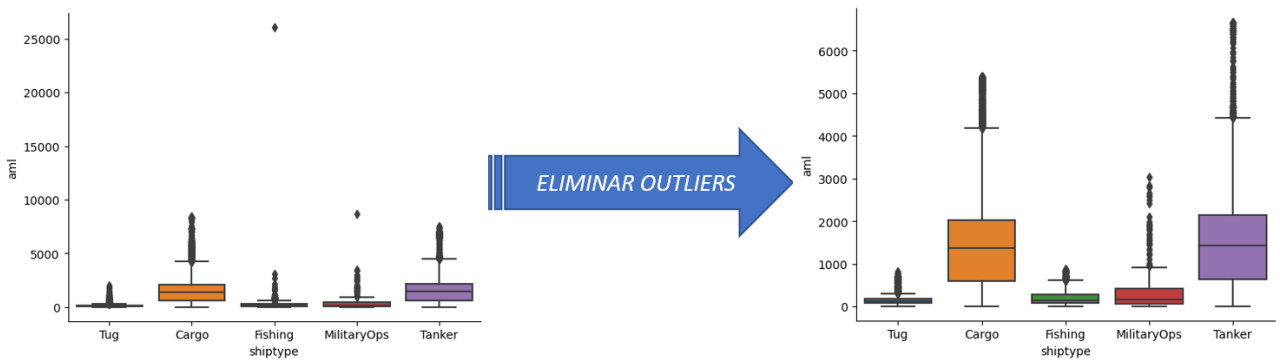


Figura 3-7. Ejemplo de eliminación de *outliers* (fuente: propia)

Para ello se han ejecutado las siguientes líneas de código, para, en este caso, obtener el percentil 0.98 de cada atributo y eliminar todas las filas de datos que no estén dentro de ese margen.

```
shiptypes = ['PETROLERO','MERCANTE','PESQUERO','REMOLCADOR', 'PASAJEROS']
atributos = df.select_dtypes(include=np.number).columns.tolist()


for m in shiptypes:
    for n in atributos:
        q = df[df['shiptype'] == m][n].quantile(0.98)
        df[(df['shiptype'] == m) & (df[n] > q)] = None
        df.loc[(df['shiptype'] == m) & (df[n] > q)] = None
df = df.dropna()
```

3.5 Experimentos realizados

Los experimentos se han dividido en tres partes. Inicialmente se han realizado experimentos utilizando únicamente los datos estáticos partiendo del código de anterior, una vez depurado. Por otro lado, se ha ejecutado el código de estadísticos dinámicos añadiendo como novedad los atributos de las celdas H3. Por último, se ha realizado un análisis global con todos los parámetros.

Cabe reseñar que para los experimentos se ha decidido eliminar las clases menos representativas: barcos de ocio y militares. Al haber tan pocos barcos de dichas clases el modelo no es incapaz de predecir correctamente una mínima parte de ellos, incluso haciendo *oversampling*. Al ser tan escasos los datos reales provocaba que los datos ficticios que se generan no fuesen suficientes. Además, muchos estudios en este ámbito ([68], [69], [70]) utilizan solamente las cinco clases más representativas.

3.5.1 Estáticos

Los experimentos realizados han estado condicionados por el calado en el análisis del calado. Durante la realización del TFG se ha observado que un parámetro que aparentemente era estático variaba en los diferentes mensajes.

El calado de un barco varía en función de la carga que lleve a bordo y del tipo de agua por la que navegue. Cuando un barco va completamente descargado, el volumen de agua que desplaza (∇) va a ser mucho menor que si va a plena carga provocando una variación en el calado del buque muy significativa. Esto provocaba en el modelo inicial unos resultados muy buenos pero que no eran reales. El motivo era el que queda reflejado en la Tabla 3-6, los atributos de la columna de la izquierda son independientes del calado, luego son iguales en un mismo barco, y los de la derecha son dependientes de él, por lo que dependerán del calado que tuviese el barco en cada mensaje. Se estaba tratando de predecir el tipo de

barco en el conjunto de prueba con los mismos barcos con los que se estaba entrenando el modelo, lo que se traducía en un incremento de acierto en la predicción de forma incorrecta. Además, la dotación de un buque puede cambiar la información de estos datos estáticos, entre ellos el calado [71].

Para aprovechar todas las muestras lo que se ha decidido es dividir los conjuntos de entrenamiento y test en base a los MMSI únicos. De esta forma en cada conjunto puede haber entradas repetidas de un mismo barco con diferentes calados, pero no en conjuntos distintos.

INDEPENDIENTES	DEPENDIENTES
len	*ldivd*
wid	*wdivd*
ldivw	*wdivd*
area	*aml*
grith	*amt*
aol	*vs*

Tabla 3-6. Dependencia del calado en los atributos estáticos

También se han eliminado todas las entradas con valores sin sentido, entre las que se destacan las siguientes:

- MMSI iguales a 0. La subdivisión de los conjuntos de prueba y entrenamiento se hace en base a este parámetro por lo que no tienen sentido valores nulos.
- $A + B = 0$ o $C + D = 0$. si la suma de dichos valores es igual a 0, significa que el transpondedor AIS no esta en ninguna parte del barco, por lo que tampoco tiene sentido, esto se puede ver en la Figura 3-1.
- Calado igual a 0. Los barcos con calados iguales a 0 se traducen en buques que no tienen volumen de carena (Figura 3-8), de nuevo incoherente.

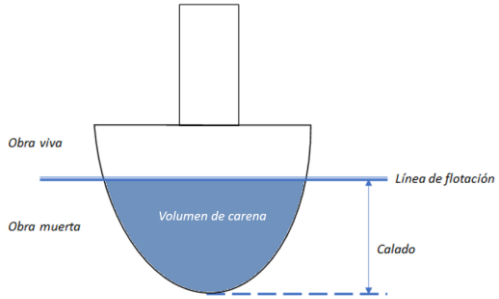


Figura 3-8. Estructura básica de un barco (fuente: propia)

Una vez depurada el conjunto de datos, se realizarán los experimentos de la Tabla 3-7 donde quedan reflejados los experimentos en base a las diferentes propuestas de optimización del código. Para todos ello se ha usado una base de datos de 75 días.

N.º Experimento	***Oversampling***	***Outliers* 0.99**	***Outliers* 0.98**
1	SI	NO	NO
2	NO	NO	NO
3	SI	SI	NO
4	NO	SI	NO
5	SI	NO	SI
6	NO	NO	SI

Tabla 3-7. Experimentos con datos estáticos

3.5.2 Dinámicos

El análisis de los datos dinámicos se ha dividido en tres partes:

Inicialmente se ha depurado el código de datos dinámicos que perfiló en el Trabajo de Fin de Grado mencionado en el apartado 2.6.1. a la vez que se desarrollaba el código para generar los parámetros estadísticos de las celdas H3. Una vez generados ambos códigos se han realizado experimentos por separado y en conjunto para medir la efectividad de las celdas H3. Además, se ha analizado el tamaño del *dataset* en base al número de días. En la Tabla 3-8 se puede ver un resumen de los experimentos realizados en la primera fase con los datos separados.

N.º Experimento	**Tamaño *Dataset***	***Oversampling***	**Dinámicos**	**Celdas H3**
7	1 día	SI	NO	SI
8	1 día	NO	NO	SI
9	1 día	SI	SI	NO
10	1 día	NO	SI	NO
11	14 días	SI	NO	SI
12	14 días	NO	NO	SI
13	14 días	SI	SI	NO
14	14 días	NO	SI	NO

Tabla 3-8. Análisis independiente con datos dinámicos

Una vez definido los datos por separado se analizarán en su conjunto (segunda fase) evaluando la posibilidad de eliminar los datos anómalos (*outliers*). Se han estudiado dos posibilidades: eliminar datos que estén fuera del percentil 99% (*outliers* 0.99) y eliminar datos que estén fuera del percentil 98% (*outliers* 0.98). La Tabla 3-9 contine un resumen de los experimentos realizados.

N.º Experimento	**Tamaño *dataset***	***Oversampling***	***Outliers* 0.99**	***Outliers* 0.98**
15	1 día	NO	NO	NO
16	1 día	SI	NO	NO
17	1 día	NO	SI	NO
18	1 día	SI	SI	NO
19	1 día	NO	NO	SI
20	1 día	SI	NO	SI
21	14 días	NO	NO	NO
22	14 días	SI	NO	NO
23	14 días	NO	SI	NO
24	14 días	SI	SI	NO
25	14 días	NO	NO	SI
26	14 días	SI	NO	SI
Estudio tamaño *dataset* (tercera fase)				
27	2 días	NO	NO	SI
28	3 días	NO	NO	SI
29	4 días	NO	NO	SI
30	5 días	NO	NO	SI
31	6 días	NO	NO	SI
32	7 días	NO	NO	SI
33	8 días	NO	NO	SI
34	9 días	NO	NO	SI
35	10 días	NO	NO	SI
36	11 días	NO	NO	SI
37	12 días	NO	NO	SI
38	13 días	NO	NO	SI

Tabla 3-9. Análisis conjunto de datos dinámicos

3.5.3 Experimentos conjunto dinámico y estático

Una vez realizados los experimentos por separado se juntarán todos los atributos con el fin de ver la repercusión que tienen las celdas H3 al modelo conjunto.

N.º Experimento	Tamaño *dataset*	Outliers 0.99	Outliers 0.98	Atributos
45	14 día	NO	SI	Todos
46	14 días	SI	NO	Todos
47	14 días	NO	NO	Todos
48	14 días	SI	NO	Óptimos (20)
49	14 días	SI	NO	Óptimos (16)

Tabla 3-10. Análisis conjunto de datos estáticos y dinámicos

4 RESULTADOS DEL TFG

En el siguiente capítulo se hará un resumen de los diferentes experimentos realizados comentando los resultados obtenidos. Inicialmente se analizarán los resultados de cada tipo de información por separado (datos estáticos y dinámicos) aplicando las posibles mejoras expuestas en el capítulo 3. También se estudiará de manera más profunda la eficacia de incorporar las celdas H3 como parámetro dinámico. Por último, se intentará definir el modelo de predicción más eficiente y óptimo.

Para agilizar la redacción de los resultados de los experimentos se hará referencia a ellos en base a la nomenclatura utilizada en las tablas expuestas en el aparatado 3.5 del capítulo anterior (Tabla 3-7, Tabla 3-8, Tabla 3-9 y Tabla 3-10). Reseñar que solo se han mencionado los experimentos más relevantes.

4.1 Métricas empleadas

Para el análisis de los datos se ha utilizado la función *classification_report* de la librería *Sklearn.* Mediante esta función se analizan una serie de parámetros que se explicarán a continuación [72]:

- ***Accuracy*** (exactitud): representa el porcentaje de datos que el modelo es capaz de acertar. La fórmula que utiliza para su cálculo es la que aparece a continuación, donde si se mide la detección de buques mercantes, TP haría referencia a los identificados correctamente como mercantes, TN a los no identificados como mercantes porque no lo son, FP y FN a los identificados erróneamente.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

  Esta métrica es desaconsejable debido a que suele llevar a engaño como se mostrará en el ejemplo siguiente: la Figura 4-1 muestra un conjunto de datos formado por los barcos de la tabla. El modelo identifica 85 barcos como mercantes, 10 como pesqueros y 5 como remolcadores. El valor del *accuracy* será del 70%, pero como se puede observar el modelo no es capaz de detectar las clases menos representadas: solo 2 de cada 5 pesqueros se detectan correctamente y solo 1 de cada 3 remolcadores. Es por ello por lo que, aunque con esta métrica los resultados obtenidos puedan ser buenos, no representan a las clases menores.

CLASES DE BARCO	Número
Mercantes	60 barcos
Pesqueros	25 barcos
Remolcadores	15 barcos

Figura 4-1. Ejemplo para analizar métricas (fuente: propia)

- ***Precision*** (precisión): es una medida que se utilizada para determinar la calidad en las tareas de clasificación, representa el número total de elementos identificados correctamente como positivos del total que han sido identificados como tal. La fórmula utilizada para su cálculo es la siguiente:

$$Precision = \frac{TP}{TP + FP}$$

  Volviendo al ejemplo planteado anteriormente (Figura 4-1) el valor de la precisión para el caso de los mercantes sería de un 70%, mientras que de los pesqueros y remolcadores sería del 100% ya que todos los barcos que han sido identificados como tal realmente lo son.

- ***Recall*** (exhaustividad): la exhaustividad se utiliza para informar sobre la cantidad de datos que el modelo es capaz de identificar. Se calcula como el cociente entre datos clasificados correctamente y los datos totales de esa clase. La fórmula sería la que aparece a continuación:

$$Recall = \frac{TP}{TP + FN}$$

  Analizando de nuevo el ejemplo de la Figura 4-1, el valor de exhaustividad para los mercantes sería del 100% (todos los mercantes son identificados), mientras que el de los pesqueros es del 40% y el de los remolcadores del 33,33%.

- ***F1-score***: es la medida que combina la precisión y la exhaustividad. Es es la óptima, ya que se utiliza en los casos en los que interesa tener en cuenta ambas métricas, a veces tener en cuenta una de las dos puede llevar a error. El valor de *f1-score* sería el más adecuado para analizar el ejemplo de la Figura 4-1. Los resultados de precisión y *recall* pueden variar mucho en función de la métrica tomada como se ha visto en los ejemplos anteriores: el *recall* para los mercantes tiene un valor del 100% lo que aparentemente es algo muy positivo, porque como ya se ha explicado no se tiene en cuenta el valor de los identificados erróneos. Por el contrario, valorando la precisión con los pesqueros y los remolcadores se obtienen valores del 100%, en este caso no tiene en cuenta todos los barcos que no han sido identificados de dichas clases. Mediante el *f1-score* se obtiene una media de dichos atributos

que se aproxima más a la realidad y permite ver de manera más rápida los resultados obtenidos.

Es por ello por lo que el *f1-score* será el parámetro que más en cuenta se tendrá en este trabajo. Adicionalmente, el informe generado proporciona unas medias aritméticas (*macro avg*) de las diferentes métricas y unas medias ponderadas (*weighted avg*) en función de la cantidad de datos.

4.2 Experimentos con conjunto de datos estáticos

Como se ha explicado en el apartado 3.5.1 lo que se ha buscado a la hora de realizar los experimentos ha sido depurar el conjunto de datos con el fin de obtener un *dataset* con muestras lo más realistas posibles, eliminando todos los datos anómalos. Una vez finalizada esta fase, se ha pasado a la realización de los seis experimentos mencionados en la Tabla 3-7 con el objetivo de definir el modelo óptimo. Inicialmente se ha analizado el hecho de eliminar el *oversampling*. La Tabla 4-1 muestra los resultados obtenidos aplicando *oversampling* sobre el conjunto de entrenamiento (1) y sin aplicarlo (2).

Los resultados obtenidos son muy similares, aunque merece la pena destacar los valores de *f1-score* de las clases menos representativas (pasajeros, pesqueros y remolcadores) cuyos valores mejoran entre un 3% y un 5%. Además, el valor de *accuracy* también mejora ligeramente y, en general, al no hacer *oversampling* no hay ningún valor que empeore. Es por esto por lo que en los experimentos siguientes no se hará *oversampling*.

EXPERIMENTO N.º 1

	precision	recall	f1-score	support
MERCANTE	0.95	0.89	0.92	35184
PASAJEROS	0.60	0.83	0.70	1326
PESQUERO	0.67	0.80	0.73	1221
PETROLERO	0.81	0.88	0.84	14067
REMOLCADOR	0.73	0.82	0.77	1845
accuracy			0.88	53643
macro avg	0.75	0.84	0.79	53643
weighted avg	0.89	0.88	0.88	53643

EXPERIMENTO N.º 2

	precision	recall	f1-score	support
MERCANTE	0.92	0.94	0.93	35184
PASAJEROS	0.76	0.71	0.74	1326
PESQUERO	0.77	0.79	0.78	1221
PETROLERO	0.88	0.82	0.85	14067
REMOLCADOR	0.80	0.80	0.80	1845
accuracy			0.90	53643
macro avg	0.83	0.81	0.82	53643
weighted avg	0.89	0.90	0.89	53643

Tabla 4-1. Comparativa de experimentos con y sin *oversampling* (fuente: propia)

Establecido que se descartará la realización de *oversampling*, se pasará a analizar cómo afecta la eliminación de datos anómalos sobre la muestra. Se han estudiado dos posibilidades: eliminar los datos que estén fuera del percentil 0.99 o los que estén fuera del percentil 0.98. La Tabla 4-2 muestra los resultados obtenidos: el experimento 4 hace referencia a los *outliers* eliminados por estar fuera del percentil 0.99 y el experimento 6 a los eliminados por estar fuera del percentil 0.98.

EXPERIMENTO N.º 4

	precision	recall	f1-score	support
MERCANTE	0.92	0.94	0.93	30592
PASAJEROS	0.78	0.74	0.76	1152
PESQUERO	0.77	0.81	0.79	1067
PETROLERO	0.87	0.81	0.84	12170
REMOLCADOR	0.82	0.86	0.84	1595
accuracy			0.90	46576
macro avg	0.83	0.83	0.83	46576
weighted avg	0.90	0.90	0.90	46576

EXPERIMENTO N.º 6

	precision	recall	f1-score	support
MERCANTE	0.92	0.94	0.93	26250
PASAJEROS	0.79	0.76	0.78	981
PESQUERO	0.78	0.83	0.80	940
PETROLERO	0.86	0.81	0.83	10578
REMOLCADOR	0.86	0.91	0.88	1357
accuracy			0.90	40106
macro avg	0.84	0.85	0.85	40106
weighted avg	0.89	0.90	0.89	40106

Tabla 4-2. Experimentos eliminando datos anómalos (fuente: propia)

Los resultados obtenidos de nuevo son muy similares, en una visión general se puede ver como los valores de *f1-score* son peores en el experimento 4, *outliers* al 0.99, pero las diferencias son mínimas.

Si bien, en el caso de los remolcadores las diferencias son más significativas pasando de un *f1-score* del 0.84 al 0.88. Las pérdidas con respecto a la muestra inicial son del 16% en el caso del percentil 0,99 y del 24% en el del percentil 0.98, pero de nuevo, volviendo al objetivo que se propuso inicialmente, que era tener un conjunto de datos lo más depurado posible, se ha decidido quedarse con los resultados obtenidos eliminando los *outliers* del 0.98 (experimento N.º 6).

Una vez definido el modelo óptimo, que sería el que ofrece los resultados del experimento 6, se pasa a estudiar la importancia de los atributos. La Figura 4-2 muestra por orden decreciente de importancia los atributos estáticos. En la imagen se puede observar cómo hay un parámetro que destaca sobre el resto que es la proporción de a sobre la eslora total (*aol*). Seguidos de este se encuentran otros seis parámetros: *ldivw, len, area, grith, b, a*. Todos ellos son relativos a la eslora y a la manga del barco, con lo que se concluye que la forma de la cubierta del barco define de forma certera el tipo de buque. Por ello se ha realizado un último experimento con los siete mejores atributos que aparece reflejado en la Tabla 4-3.

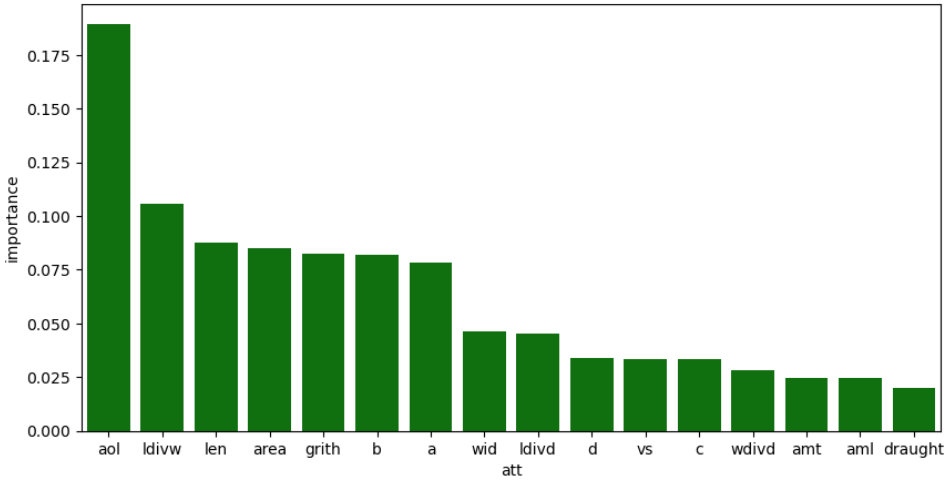


Figura 4-2. Importancia atributos modelo de datos estáticos (fuente: propia)

EXPERIMENTO CON LOS MEJORES ATRIBUTOS

	precision	recall	f1-score	support
MERCANTE	0.92	0.93	0.93	21386
PASAJEROS	0.59	0.59	0.59	679
PESQUERO	0.74	0.76	0.75	797
PETROLERO	0.84	0.81	0.83	8023
REMOLCADOR	0.87	0.93	0.90	1105
accuracy			0.89	31990
macro avg	0.79	0.80	0.80	31990
weighted avg	0.89	0.89	0.89	31990

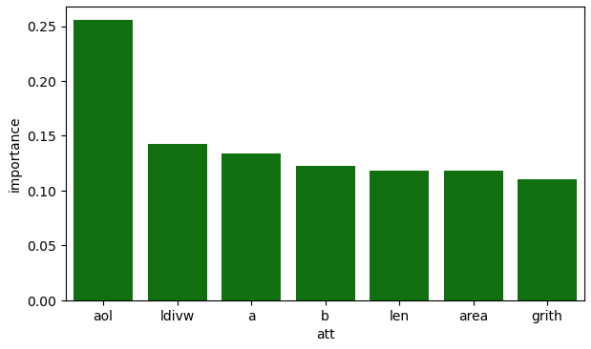


Tabla 4-3. Experimento con los 6 mejores atributos (fuente: propia)

La Tabla 4-3 muestra los resultados obtenidos al utilizar solamente los atributos más significativos, como se puede ver el valor del *accuracy* no varía significativamente, en cambio el valor de *f1-score* sí que empeora bastante. Pasa en el experimento 6 de un valor de 0.78 a un valor de 0.59 para la clase de pasajeros. En la única clase en la que se puede observar una mejoría es en la de remolcadores donde mejora de un 0.88 a un 0.90. Como se puede observar las pérdidas son mayores que las ganancias por eso el modelo óptimo contará con todos los atributos y quedará definido según la Tabla 4-4.

DEFINICIÓN DEL MODELO	
N.º de árboles	100 árboles
División datos	75/25
Oversampling	NO
Undersampling	NO
MinMaxScaler	SI
Outliers	PERCENTIL 0.98
Atributos	TODOS

Tabla 4-4. Definición del modelo para el conjunto de datos estáticos

4.3 Experimentos con conjunto de datos dinámicos

Los datos dinámicos se han dividido inicialmente en dos subgrupos para ver su influencia sobre el modelo. En la Tabla 4-5 se pueden ver los experimentos números 9, 10, 13 y 14, en los que se han utilizado datos dinámicos sin celdas H3. En los experimentos 9 y 13 se realiza *oversampling* en los datos de entrenamiento. Finalmente, en los experimentos 9 y 10 se utiliza un dataset de 1 día mientras que en los otros dos uno de 14 días.

EXPERIMENTO N.º 9

	precision	recall	f1-score	support
MERCANTE	0.72	0.66	0.69	2662
PASAJEROS	0.42	0.66	0.52	307
PESQUERO	0.59	0.72	0.65	534
PETROLERO	0.40	0.37	0.39	1098
REMOLCADOR	0.18	0.21	0.20	42
accuracy			0.60	4643
macro avg	0.46	0.52	0.49	4643
weighted avg	0.60	0.60	0.60	4643

EXPERIMENTO N.º 10

	precision	recall	f1-score	support
MERCANTE	0.65	0.94	0.77	2662
PASAJEROS	0.73	0.43	0.54	307
PESQUERO	0.73	0.61	0.67	534
PETROLERO	0.53	0.09	0.16	1098
REMOLCADOR	1.00	0.02	0.05	42
accuracy			0.66	4643
macro avg	0.73	0.42	0.44	4643
weighted avg	0.64	0.66	0.59	4643

EXPERIMENTO N.º 13

	precision	recall	f1-score	support
MERCANTE	0.71	0.66	0.68	37504
PASAJEROS	0.50	0.68	0.58	4432
PESQUERO	0.62	0.73	0.67	7431
PETROLERO	0.43	0.38	0.40	16150
REMOLCADOR	0.37	0.68	0.48	2098
accuracy			0.60	67615
macro avg	0.53	0.63	0.56	67615
weighted avg	0.61	0.60	0.60	67615

EXPERIMENTO N.º 14

	precision	recall	f1-score	support
MERCANTE	0.65	0.94	0.76	37504
PASAJEROS	0.82	0.49	0.61	4432
PESQUERO	0.76	0.65	0.70	7431
PETROLERO	0.55	0.10	0.17	16150
REMOLCADOR	0.69	0.39	0.50	2098
accuracy			0.66	67615
macro avg	0.69	0.52	0.55	67615
weighted avg	0.65	0.66	0.60	67615

Tabla 4-5. Experimentos datos dinámicos sin celdas H3 (fuente: propia)

La Tabla 4-6 muestra los experimentos 7, 8, 11 y 12 en los cuales se utilizan solo los parámetros relativos a las celdas H3. Siguiendo el mismo criterio que en los anteriores, a los impares se aplica *oversampling* y los dos primeros trabajan con un *dataset* de un día y los dos últimos de un *dataset* de 14 días.

EXPERIMENTO N.º 7

	precision	recall	f1-score	support
MERCANTE	0.65	0.66	0.66	2722
PASAJEROS	0.47	0.48	0.47	331
PESQUERO	0.47	0.58	0.52	505
PETROLERO	0.31	0.26	0.28	1212
REMOLCADOR	0.13	0.14	0.14	49
accuracy			0.53	4819
macro avg	0.40	0.42	0.41	4819
weighted avg	0.53	0.53	0.53	4819

EXPERIMENTO N.º 8

	precision	recall	f1-score	support
MERCANTE	0.63	0.83	0.72	2722
PASAJEROS	0.70	0.44	0.54	331
PESQUERO	0.61	0.53	0.57	505
PETROLERO	0.29	0.13	0.18	1212
REMOLCADOR	0.45	0.10	0.17	49
accuracy			0.59	4819
macro avg	0.53	0.41	0.43	4819
weighted avg	0.54	0.59	0.55	4819

EXPERIMENTO N.º 11

	precision	recall	f1-score	support
MERCANTE	0.66	0.67	0.66	37171
PASAJEROS	0.49	0.49	0.49	4520
PESQUERO	0.49	0.60	0.54	7146
PETROLERO	0.32	0.25	0.28	16186
REMOLCADOR	0.35	0.53	0.42	2039
accuracy			0.55	67062
macro avg	0.46	0.51	0.48	67062
weighted avg	0.54	0.55	0.54	67062

EXPERIMENTO N.º 12

	precision	recall	f1-score	support
MERCANTE	0.63	0.85	0.72	37171
PASAJEROS	0.70	0.43	0.53	4520
PESQUERO	0.60	0.52	0.56	7146
PETROLERO	0.32	0.13	0.18	16186
REMOLCADOR	0.51	0.39	0.44	2039
accuracy			0.60	67062
macro avg	0.55	0.46	0.49	67062
weighted avg	0.56	0.60	0.56	67062

Tabla 4-6. Experimentos basados exclusivamente en celdas H3 (fuente: propia)

Como conclusiones de estos ocho experimentos iniciales se puede destacar que los atributos dinámicos sin celdas H3 tienen mejores resultados que los referentes a los estadísticos de celdas H3. Además, realizar *oversampling* empeora los resultados significativamente en todos los casos. La clase de buque peor identificada es la de remolcadores. Esto es causado porque la representación de la muestra en el *dataset* es muy escasa. El tipo de barco mejor clasificado es el mercante con un *f1-score* de 0.77 en el mejor de los casos (experimento 10).

Por otro lado, se ha utilizado el algoritmo de *kNN* para aplicar sobre las celdas H3, como se explicó en el apartado 3.3.2, pero finalmente se ha desechado esta opción. Los procesos de ejecución de este modelo eran muy lentos, debido al gran tamaño del *dataset*, que estaba formado por todas las celdas H3 por las que los barcos habían pasado. Al no ser parámetros estadísticos, el número de datos crecía exponencialmente, llegando a generar bloqueos en el ordenador. Aun así, los datos obtenidos han sido buenos.

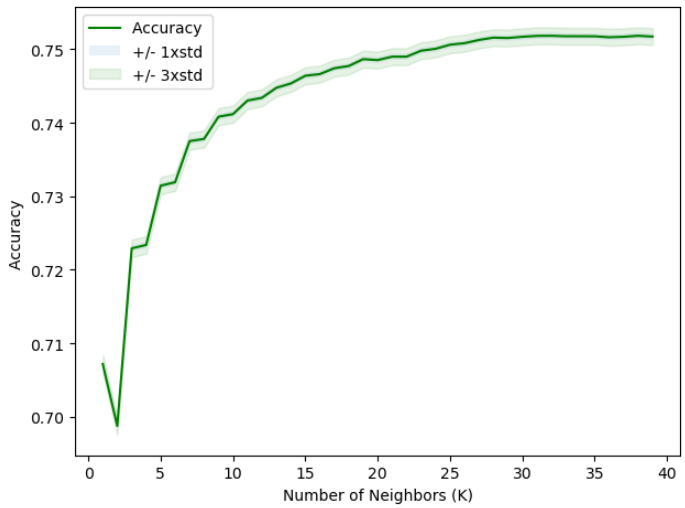


Figura 4-3. Optimización valor de *k* (fuente: propia)

Inicialmente se calculó el k optimo. La Figura 4-3 muestra cómo en función del valor de *k* (eje x) se estabiliza la exactitud (*accuracy*) de la predicción. En $k = 25$ se estabiliza, pero el valor óptimo se obtiene para $k = 38$. Para su cálculo no solo se tiene en cuenta la exactitud, sino que el algoritmo también analiza otros parámetros como el tiempo de ejecución. A partir de esto se realizaron las pruebas que aparecen a continuación.

PRUEBA CON *OVERSAMPLING*

	precision	recall	f1-score	support
MERCANTE	0.87	0.58	0.70	376257
PASAJEROS	0.37	0.82	0.51	28733
PESQUERO	0.71	0.92	0.80	76337
PETROLERO	0.50	0.69	0.58	147865
REMOLCADOR	0.24	0.89	0.38	4291
accuracy			0.66	633483
macro avg	0.54	0.78	0.59	633483
weighted avg	0.74	0.66	0.67	633483

PRUEBA SIN *OVERSAMPLING*

	precision	recall	f1-score	support
MERCANTE	0.75	0.89	0.81	376257
PASAJEROS	0.66	0.48	0.56	28733
PESQUERO	0.82	0.81	0.82	76337
PETROLERO	0.64	0.38	0.47	147865
REMOLCADOR	0.59	0.39	0.47	4291
accuracy			0.74	633483
macro avg	0.69	0.59	0.63	633483
weighted avg	0.73	0.74	0.72	633483

Tabla 4-7. Pruebas del algoritmo *kNN* (fuente: propia)

Como se puede observar los resultados obtenidos, el de 0.82 en pesqueros es el mejor de los casos. Dichos resultados no son eficientes ni precisos, ya que en el modelo definido no tiene en cuenta la frecuencia de las celdas por las que navega un determinado barco. Como consecuencia y debido también a los largos tiempos de ejecución mencionados anteriormente se ha decidido descartar este modelo.

Una vez estudiados los parámetros por separado se pasan a analizar todos los atributos dinámicos en conjunto. Tras los resultados obtenidos en la Tabla 4-5 y en la Tabla 4-6 se ha decidido descartar el *oversampling* en los datos de entrenamiento, con el fin de trabajar exclusivamente con datos reales que son los que ofrecen mejores resultados. Inicialmente se han realizado experimentos para *dataset* de dos tipos: *dataset* de tamaño uno (el cual incluye los datos AIS de un solo día) y *dataset* de tamaño 14 (el cual incluye los datos AIS de catorce días). La Tabla 4-8 refleja los experimentos realizados para ambos tamaños sin realizar *oversampling* a los datos de entrenamiento.

EXPERIMENTO N.º 15

	precision	recall	f1-score	support
MERCANTE	0.66	0.94	0.78	2662
PASAJEROS	0.76	0.54	0.63	307
PESQUERO	0.77	0.63	0.70	534
PETROLERO	0.53	0.09	0.16	1098
REMOLCADOR	1.00	0.02	0.05	42
accuracy			0.67	4643
macro avg	0.75	0.45	0.46	4643
weighted avg	0.65	0.67	0.60	4643

EXPERIMENTO N.º 21

	precision	recall	f1-score	support
MERCANTE	0.65	0.94	0.77	37504
PASAJEROS	0.86	0.56	0.68	4432
PESQUERO	0.80	0.65	0.72	7431
PETROLERO	0.57	0.11	0.19	16150
REMOLCADOR	0.70	0.48	0.57	2098
accuracy			0.67	67615
macro avg	0.72	0.55	0.59	67615
weighted avg	0.66	0.67	0.61	67615

Tabla 4-8. Experimentos iniciales con todos los atributos dinámicos (fuente: propia)

En ambos experimentos se puede observar cómo los resultados mejoran respecto a los realizados solo con datos dinámicos sin celdas H3. La exactitud es muy similar (solo dinámicos de un valor de 0.66 y en dinámicos y celdas H3 de 0.67) pero los barcos con peores resultados mejoran significativamente. El caso de los remolcadores, en el experimento 21 tiene un *f1-score* de 0.57 cuando en el experimento 10 era de 0.05 con un valor de precisión de 1.00, como se explicó en el apartado 4.1 una precisión tan elevada y un *recall* tan bajo 0.02 se traduce en que todo lo que el modelo identificaba como remolcador es un barco de esa clase pero solo era capaz de identificar un 2%, mientras que sumándole al modelo la

utilización de celdas H3 se consigue que identifique un 48% de los barcos, de los cuales un 70% son correctos.

Además, se pasará a evaluar la posibilidad de eliminar datos anómalos (*outliers*) explicados en el apartado 3.4.2. Los dos posibles casos que se han analizado han sido:

- ***Outliers 0.99**, donde se eliminarían todas aquellas muestras que no estuviesen dentro del percentil 0.99.*
- ***Outliers* 0.98**, donde se eliminarían aquellas que no perteneciesen al percentil 0.98.

Inicialmente se han realizado experimentos para *dataset* de dos tipos: *dataset* de tamaño uno (el cual incluye los datos AIS de un solo día) y *dataset* de tamaño 14 (el cual incluye los datos AIS de catorce días). La Tabla 4-9 refleja los experimentos realizados con el *dataset* de un día y la Tabla 4-10 representa los resultados de los experimentos realizados con el *dataset* de 14 días. El experimento que se encuentra en la parte izquierda de la tabla (17 y 23 respectivamente) corresponde con el experimento en el que se eliminan los *outliers* 0.99 siendo los de la columna de la derecha (19 y 25) los restantes (*outliers* 0.98).

EXPERIMENTO N.º 17

	precision	recall	f1-score	support
MERCANTE	0.70	0.96	0.81	1229
PASAJEROS	0.75	0.60	0.67	120
PESQUERO	0.86	0.77	0.81	233
PETROLERO	0.51	0.13	0.20	504
REMOLCADOR	1.00	0.18	0.30	17
accuracy			0.71	2103
macro avg	0.77	0.53	0.56	2103
weighted avg	0.68	0.71	0.65	2103

EXPERIMENTO N.º 19

	precision	recall	f1-score	support
MERCANTE	0.70	0.97	0.81	551
PASAJEROS	0.91	0.61	0.73	66
PESQUERO	0.84	0.88	0.86	89
PETROLERO	0.51	0.11	0.18	242
REMOLCADOR	0.00	0.00	0.00	3
accuracy			0.71	951
macro avg	0.59	0.51	0.51	951
weighted avg	0.68	0.71	0.65	951

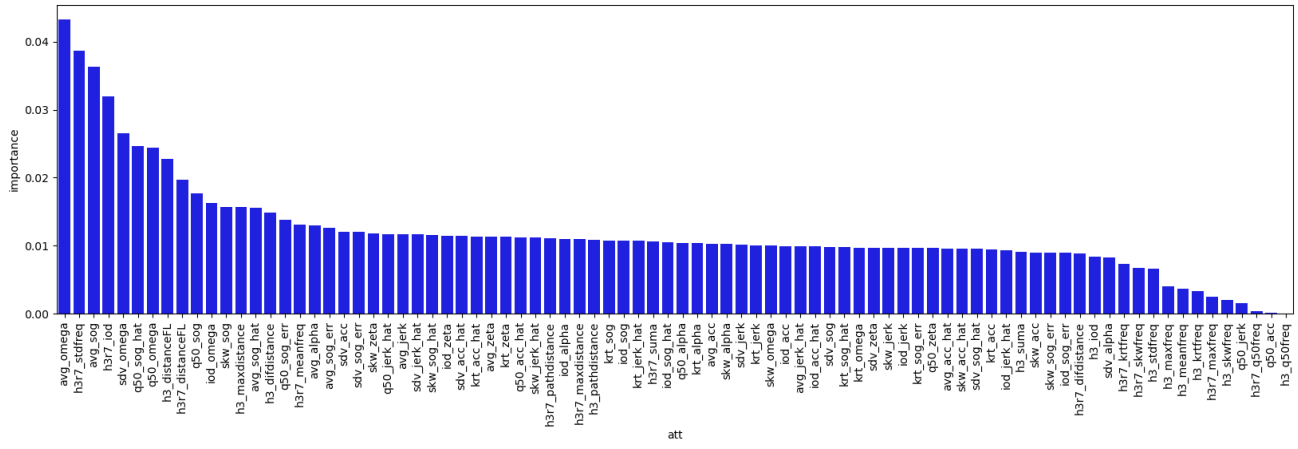


Tabla 4-9. Experimentos *dataset* tamaño 1 (fuente: propia)

EXPERIMENTO N.º 23

	precision	recall	f1-score	support
MERCANTE	0.69	0.96	0.80	16594
PASAJEROS	0.89	0.66	0.76	2048
PESQUERO	0.86	0.82	0.84	3244
PETROLERO	0.59	0.13	0.21	7301
REMOLCADOR	0.86	0.77	0.81	911
accuracy			0.72	30098
macro avg	0.78	0.67	0.69	30098
weighted avg	0.70	0.72	0.66	30098

EXPERIMENTO N.º 25

	precision	recall	f1-score	support
MERCANTE	0.71	0.95	0.81	7687
PASAJEROS	0.87	0.72	0.79	862
PESQUERO	0.89	0.87	0.88	1466
PETROLERO	0.55	0.15	0.23	3291
REMOLCADOR	0.91	0.86	0.88	419
accuracy			0.73	13725
macro avg	0.78	0.71	0.72	13725
weighted avg	0.71	0.73	0.68	13725

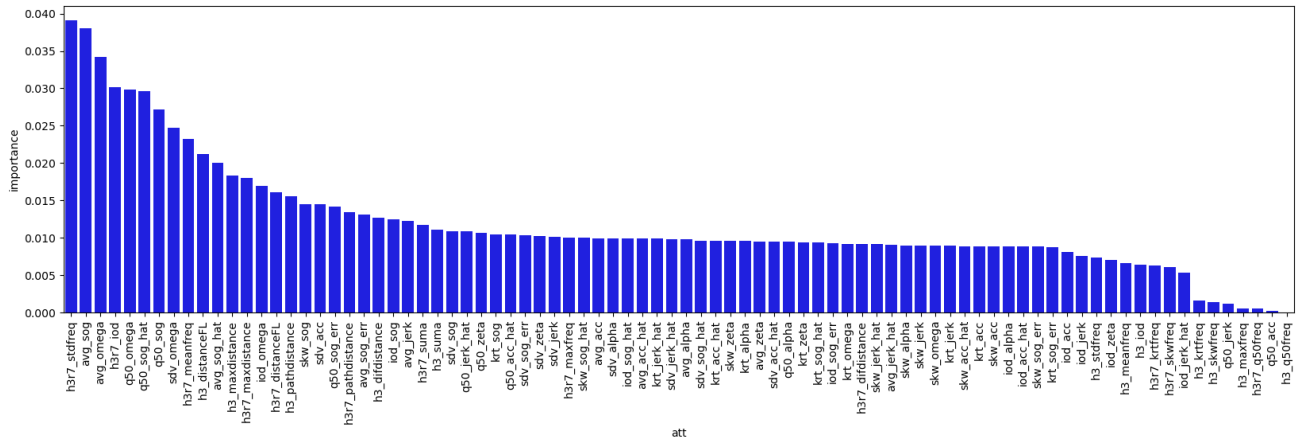


Tabla 4-10. Experimentos *dataset* tamaño 14 (fuente: propia)

Los cuatro experimentos muestran resultados similares, lo que se traduce en que un aumento del tamaño del *dataset* aparentemente no mejora significativamente en el valor de *accuracy*. Pero si es cierto que en las clases menos representadas se obtienen resultados mejores, los remolcadores mejoran muy significativamente su *f1-score* pasando de 0 a 0.88 en los experimentos 19 y 25 respectivamente.

Por otro lado, los resultados eliminando las muestras fuera del percentil 0.98 también son mejores que eliminando las del 0.99 y ambas superan los resultados de los experimentos 15 y 21 (sin eliminar datos anómalos). En contrapartida la eliminación de datos anómalos hace que se pierda una gran parte de la muestra. En el peor de los casos, *outliers* 0.98, la muestra de prueba pasa de 67615 datos a 13725 lo que se traduce en que queda reducida a un 20% de la muestra total inicial. Esto puede ser perjudicial, pero valorando el objetivo inicial del TFG, se busca tener un conjunto de datos depurado y fiable que permita definir de la mejor forma posible los comportamientos de las diferentes clases de barcos y eliminando los datos con comportamientos exageradamente diferentes a los de su clase se consigue dicho objetivo.

Pasando a evaluar el orden de importancia de los atributos, se puede observar en ambas tablas que el orden cambia. Cabe reseñar que se han tomado los órdenes de importancia de los resultados óptimos para cada uno de los casos, que en ambos casos ha sido eliminando *outliers* del percentil 0.98. Como se puede ver en la Figura 4-4, de los ocho atributos más importantes para cada caso solo se diferencian en uno: *q50_sog* y *h3_distanceFL*, el resto de los atributos son iguales. Si bien la dispersión en importancia en el caso 2 es mayor que en el 1.

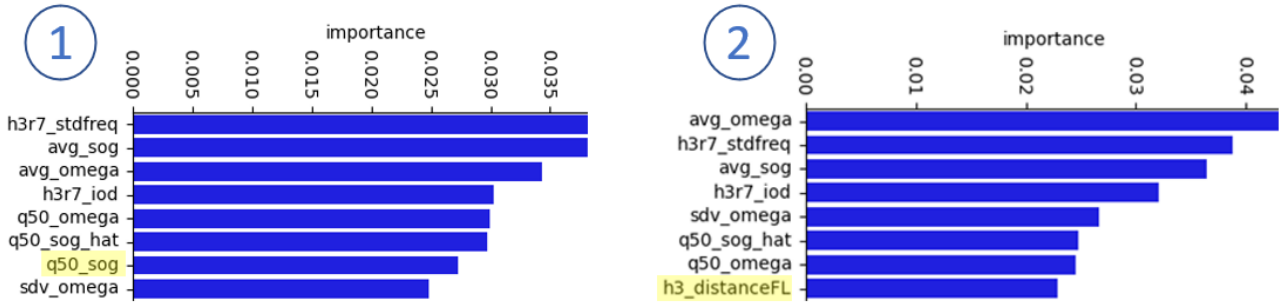


Figura 4-4. Atributos más importantes para cada caso (fuente: propia)

Una vez definido el modelo en cuanto a datos anómalos (eliminar *outliers* del percentil 0.98) se han analizado las diferentes métricas en función del tamaño del dataset. Desde 1 día hasta un dataset de 14 días. En la Figura 4-5 se puede observar la evolución de los parámetros (*f1-score*, *accuracy*) en función del tipo de barco y del número de días.

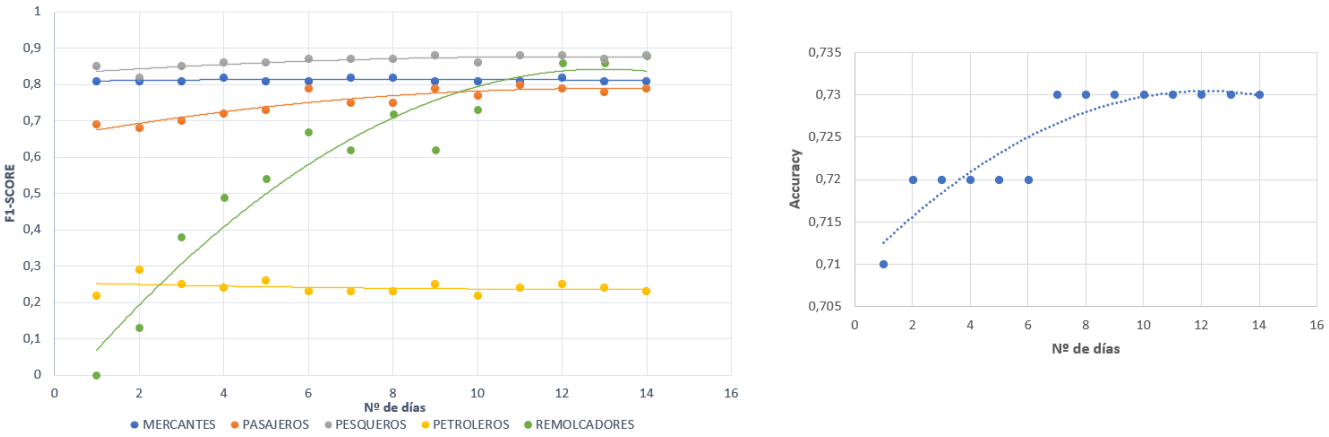


Figura 4-5. Evaluación del tamaño del *dataset* (fuente: propia)

Se puede observar que el comportamiento de todos los barcos es parecido, y que aumentar el tamaño del dataset no afecta significativamente. Hay tipos de barco con patrones más definidos como los pesqueros o mercantes donde los porcentajes de identificación se mantienen siempre elevados (en torno a un 90%). La excepción la constituyen los remolcadores y los barcos de pasajeros, a quienes aumentar el tamaño del dataset sí que ayuda en la identificación, sobre todo en los primeros. Los remolcadores pasan de un nivel de f1 despreciable hasta llegar casi al 90%, en el caso de los de pasajeros la mejora es menor, pero también se puede apreciar. En cuanto a los petroleros, el aumento del tamaño del dataset no afecta a la ayuda de su identificación por lo que se puede afirmar que esta clase de barcos es de difícil identificación solo con parámetros dinámicos. En cuanto al valor de *accuracy*, se puede observar una mejoría muy lenta en el tiempo, pero una evolución positiva, luego un aumento del tamaño del dataset siempre ayudará a una mayor eficacia del modelo.

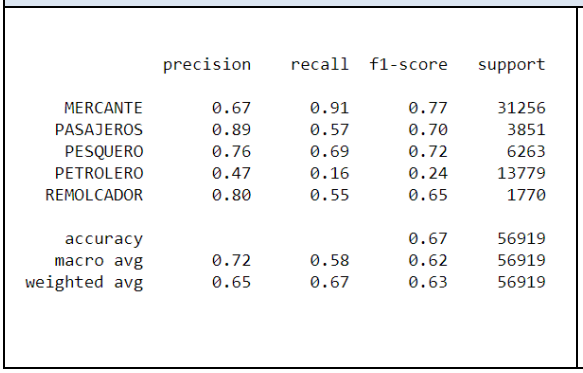
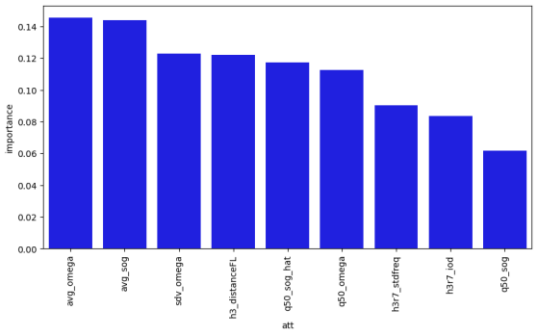
Como conclusiones principales se ha determinado que las ventajas de aumentar el tamaño del dataset se centran en conseguir un mayor nivel de confianza en la clasificación de los remolcadores y barcos de pasajeros, además de una mejora genérica lenta de todos los parámetros.

Por ello el modelo definido óptimo para los datos dinámicos es el que se muestra a continuación:

- N.º de árboles: **100**.
- Se utilizará ***MinMaxScaler***.
- División de datos: **75/25**.
- **No** se utilizará ni ***oversampling*** ni ***undersampling***.
- Se eliminarán ***outliers*** **del percentil 0.98**.

- Tamaño del dataset: **14 días**.
- **Atributos**: *h3r7_stdfreq, avg_sog, avg_omega, h3r7_iod, q50_omega, q50_sog_hat, q50_sog, sdv_omega* y *h3_distanceFL.*

Se ha realizado un último experimento para analizar si es viable desechar los atributos menos significativos. La Tabla 4-11 muestra los resultados obtenidos: se observa que empeoran los resultados, tanto los valores de *f1-score* como de accuracy, por lo que se ha decidido no descartar ningún atributo dinámico.

EXPERIMENTO CON LOS MEJORES ATRIBUTOS	
	

	precision	recall	f1-score	support
MERCANTE	0.67	0.91	0.77	31256
PASAJEROS	0.89	0.57	0.70	3851
PESQUERO	0.76	0.69	0.72	6263
PETROLERO	0.47	0.16	0.24	13779
REMOLCADOR	0.80	0.55	0.65	1770
accuracy			0.67	56919
macro avg	0.72	0.58	0.62	56919
weighted avg	0.65	0.67	0.63	56919

Tabla 4-11. Experimento con los 9 mejores atributos (fuente: propia)

En resumen, el modelo final definido es el que se muestra en la Tabla 4-12.

DEFINICIÓN DEL MODELO	
N.º de árboles	100 árboles
División datos	75/25
Oversampling	NO
Undersampling	NO
MinMaxScaler	SI
Outliers	PERCENTIL 0.98
Dataset	14 DÍAS
Atributos	TODOS

Tabla 4-12. Definición del modelo para el conjunto de datos dinámicos (fuente: propia)

4.4 Experimentos con conjuntos de datos estáticos y dinámicos

Para el análisis con todos los atributos se ha partido de los modelos predefinidos tanto para datos estáticos como dinámicos, mostrados en la Figura 4-6. Se puede observar que todos los parámetros son iguales a diferencia del tamaño del dataset, en el caso de datos estáticos alimentado con una base de datos de 60 días frente al de dinámicos, de 14 días.

DEFINICIÓN DEL MODELO	
N.º de árboles	100 árboles
División datos	75/25
Oversampling	NO
Undersampling	NO
MinMaxScaler	SI
Outliers	PERCENTIL 0.98
Atributos	TODOS

DEFINICIÓN DEL MODELO	
N.º de árboles	100 árboles
División datos	75/25
Oversampling	NO
Undersampling	NO
MinMaxScaler	SI
Outliers	PERCENTIL 0.98
Dataset	14 DÍAS
Atributos	TODOS

Figura 4-6. Definición modelos datos estáticos y dinámicos (fuente: propia)

Por ello los experimentos que se realizarán a continuación serán para un dataset de 14 días. La Tabla 4-13 muestra la comparativa de experimentos estáticos (izquierda), dinámicos (derecha) y mixto de 14 días.

EXPERIMENTO N.º 6

	precision	recall	f1-score	support
MERCANTE	0.92	0.94	0.93	26250
PASAJEROS	0.79	0.76	0.78	981
PESQUERO	0.78	0.83	0.80	940
PETROLERO	0.86	0.81	0.83	10578
REMOLCADOR	0.86	0.91	0.88	1357
accuracy			0.90	40106
macro avg	0.84	0.85	0.85	40106
weighted avg	0.89	0.90	0.89	40106

EXPERIMENTO N.º 25

	precision	recall	f1-score	support
MERCANTE	0.71	0.95	0.81	7687
PASAJEROS	0.87	0.72	0.79	862
PESQUERO	0.89	0.87	0.88	1466
PETROLERO	0.55	0.15	0.23	3291
REMOLCADOR	0.91	0.86	0.88	419
accuracy			0.73	13725
macro avg	0.78	0.71	0.72	13725
weighted avg	0.71	0.73	0.68	13725

EXPERIMENTO N.º 45

	precision	recall	f1-score	support
MERCANTE	0.93	0.94	0.93	11738
PASAJEROS	0.95	0.90	0.92	769
PESQUERO	0.95	0.98	0.96	697
PETROLERO	0.85	0.83	0.84	4728
REMOLCADOR	0.98	0.95	0.96	268
accuracy			0.91	18200
macro avg	0.93	0.92	0.92	18200
weighted avg	0.91	0.91	0.91	18200

Tabla 4-13. Experimentos *dataset* de 14 días (fuente: propia)

Como se puede observar, los resultados obtenidos son mejores que los obtenidos solamente con parámetros estáticos. En particular, mejoran sustancialmente los valores de *f1-score* de pasajeros, pesqueros y de los remolcadores pasando de 0.78, 0.80 y 0.88 respectivamente a 0.93 para pasajeros y remolcadores y 0.92 para pesqueros. También se puede observar que el número de datos respecto a estáticos disminuye sustancialmente pasando de un total de 40000 muestras a 18200 en el conjunto de test. Esto puede ser debido a varios motivos: el filtro que se realiza de datos cinemáticos que afecte significativamente a los estáticos y la eliminación de *outliers*. Para comprobar esto se ha decidido realizar experimentos sin *outliers* y con *outliers* al 0.99 con el fin de encontrar el modelo óptimo para todo el conjunto de resultados.

La Figura 4-7 muestra una comparativa entre los tres posibles casos: *outliers* 0.98, *outliers* 0.99 y sin *outliers*. Se puede observar que los mejores resultados se obtienen en el experimento 46 (*outliers* 0.99) donde disminuyen ligeramente los valores de *f1-score* mencionados anteriormente, pero en contra partida se consigue mejorar el valor de *f1-score* de los petroleros del 0.84 al 0.86. Es por ello por lo que,

para definir el modelo con los datos mixtos, la opción óptima sería con eliminando *outliers* al 0.99 (experimento N.º 46).

EXPERIMENTO N.º 45

	precision	recall	f1-score	support
MERCANTE	0.93	0.94	0.93	11738
PASAJEROS	0.95	0.90	0.92	769
PESQUERO	0.95	0.98	0.96	697
PETROLERO	0.85	0.83	0.84	4728
REMOLCADOR	0.98	0.95	0.96	268
accuracy			0.91	18200
macro avg	0.93	0.92	0.92	18200
weighted avg	0.91	0.91	0.91	18200

EXPERIMENTO N.º 46

	precision	recall	f1-score	support
MERCANTE	0.93	0.95	0.94	31392
PASAJEROS	0.93	0.87	0.90	2163
PESQUERO	0.93	0.97	0.95	1988
PETROLERO	0.88	0.84	0.86	12673
REMOLCADOR	0.96	0.88	0.92	767
accuracy			0.92	48983
macro avg	0.92	0.90	0.91	48983
weighted avg	0.92	0.92	0.92	48983

EXPERIMENTO N.º 47

	precision	recall	f1-score	support
MERCANTE	0.93	0.95	0.94	85060
PASAJEROS	0.87	0.85	0.86	5718
PESQUERO	0.90	0.93	0.92	5361
PETROLERO	0.91	0.85	0.88	34404
REMOLCADOR	0.83	0.72	0.77	2160
accuracy			0.92	132703
macro avg	0.89	0.86	0.87	132703
weighted avg	0.92	0.92	0.92	132703

Figura 4-7. Experimentos conjunto de datos mixto (fuente: propia)

La Figura 4-8 muestra una gráfica con el orden de importancia de los atributos del experimento 46, se puede ver que prevalecen los atributos estáticos como más importantes. Sobre todos ellos el atributo que mejor ayuda a definir el tipo de buque es el *aol* (proporción de *a* sobre la eslora total). Y entre los dinámicos destacan la media ponderada de la velocidad y la desviación estándar de la frecuencia de las celdas H3 de nivel 9.

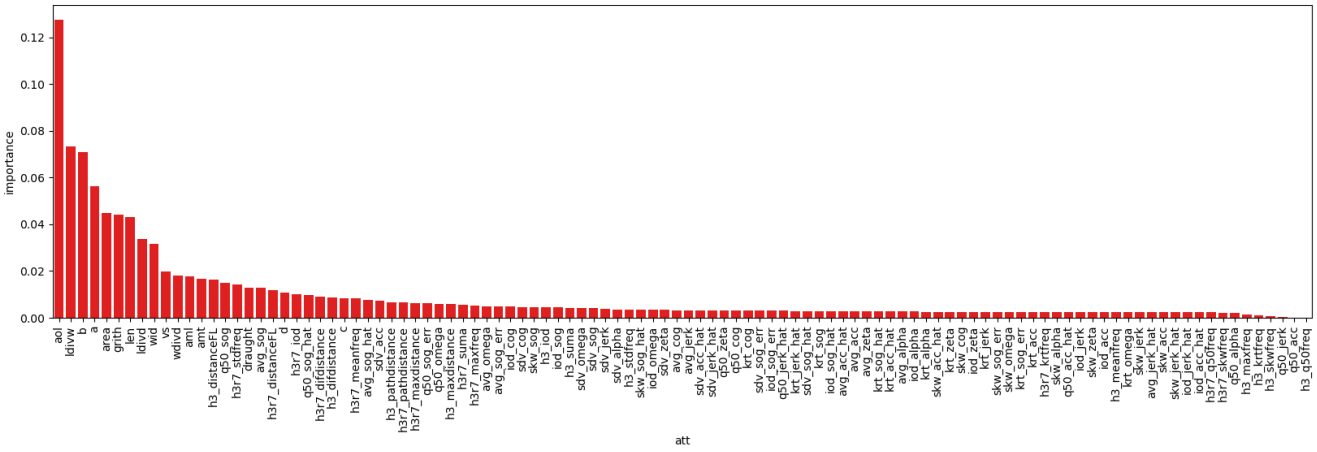
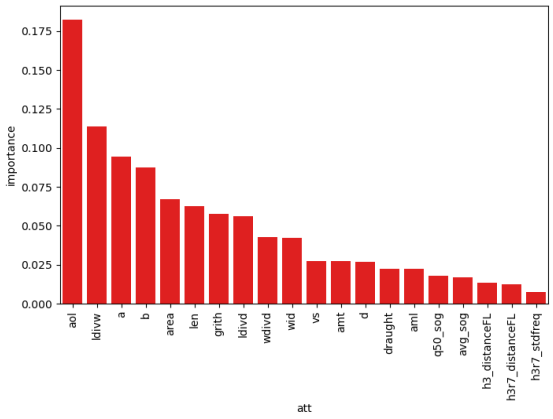


Figura 4-8. Importancia atributos experimento 46 (fuente: propia)

Con todo ello, se ha decidido realizar dos últimos experimentos: uno con los 20 atributos óptimos (experimento 48) de la Figura 4-8, que serían los que nombran a continuación: *aol*, *ldivw*, *b*, *a*, *area*, *grith*, *len*, *ldivd*, *wid*, *vs*, *wdivd*, *aml*, *amt*, *h3_distanceFL*, *q50_sog*, *h3r7_stdfreq*, *draught*, *avg_sog* y *h3r7_distanceFL*, *d*. Y otro, con los mejores atributos definidos en los casos independientes, que serían los definidos como óptimos en los experimentos de la Tabla 4-3 y Tabla 4-11 (aol, *ldivw*, *len*, *area*, *grith*, *b*, *a*, *h3r7_stdfreq*, *avg_sog*, *avg_omega*, *h3r7_iod*, *q50_omega*, *q50_sog_hat*, *q50_sog*, *sdv_omega* y *h3_distanceFL)*

EXPERIMENTO N.º 48

	precision	recall	f1-score	support
MERCANTE	0.93	0.95	0.94	71382
PASAJEROS	0.89	0.86	0.87	4964
PESQUERO	0.90	0.92	0.91	4534
PETROLERO	0.90	0.86	0.88	29661
REMOLCADOR	0.80	0.82	0.81	1856
accuracy			0.92	112397
macro avg	0.89	0.88	0.88	112397
weighted avg	0.92	0.92	0.92	112397



EXPERIMENTO N.º 49

	precision	recall	f1-score	support
MERCANTE	0.93	0.94	0.94	73540
PASAJEROS	0.81	0.81	0.81	5002
PESQUERO	0.84	0.91	0.87	4656
PETROLERO	0.90	0.86	0.88	30361
REMOLCADOR	0.83	0.77	0.80	1891
accuracy			0.91	115450
macro avg	0.86	0.86	0.86	115450
weighted avg	0.91	0.91	0.91	115450

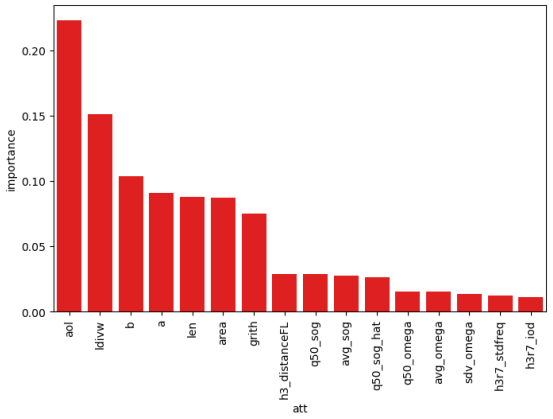


Tabla 4-14. Experimentos con atributos óptimos (fuente: propia)

Como se puede ver en la Tabla 4-14 los resultados obtenidos en el experimento 48 son muy similares a los obtenidos en el experimento 46. La principal diferencia se obtiene en los remolcadores, donde el valor de *f1-score* disminuye significativamente de 0.92 a 0.81, el resto de los valores permanecen muy igualados. Por otro lado, los resultados obtenidos con los atributos óptimos definidos en los modelos independientes son peores, a excepción de los mercantes y petroleros que se mantienen. La Figura 4-9 muestra la comparativa entre 16 atributos, 20 atributos y todos los atributos. Se puede ver cómo afecta fundamentalmente a las clases de pasajeros y de remolcadores. Esto confirma lo que se ha ido mencionando durante todo el trabajo, los parámetros cinemáticos ayudan a definir los patrones de estas dos clases de barco mientras que para el resto solo con datos estáticos se consiguen buenos resultados en las predicciones (sobre todo las clases de mercantes y de pesqueros).

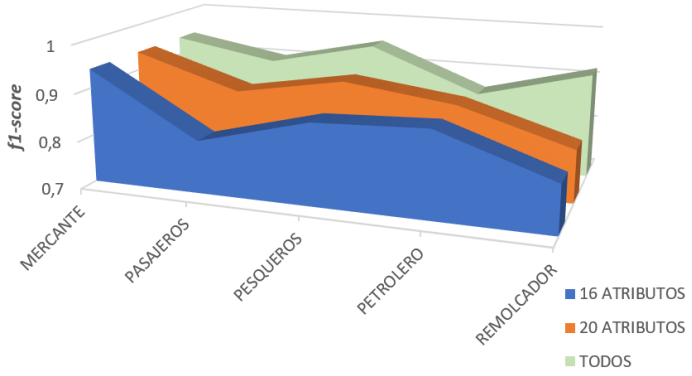


Figura 4-9. Comparativa *f1-score* atributos óptimos (fuente: propia)

Por todo ello, el modelo definido cuando trabaja con los datos en conjunto quedaría reflejado en Tabla 4-15. Se ha decidido optar por el experimento n. º 48, los resultados son muy similares al resto y tiene la ventaja de contar con parámetros dinámicos que ayudan a definir las dos clases de buque mencionadas anteriormente. De los 20 parámetros que definen el modelo quedarían un 75% de parámetros estáticos y un 25% de dinámicos (un 15% relativos a celdas H3 y un 10% a cinemáticos).

DEFINICIÓN DEL MODELO	
N.º de árboles	100 árboles
División datos	75/25
Oversampling	NO
Undersampling	NO
MinMaxScaler	SI
Outliers	PERCENTIL 0.99
Dataset	14 DÍAS
Atributos	20 ÓPTIMOS

Tabla 4-15. Definición del modelo para el conjunto de datos mixto

5 CONCLUSIONES Y LÍNEAS FUTURAS

En este capítulo se hará un resumen de las conclusiones obtenidas en base a los objetivos marcados en el capítulo 1. Además se plantearán posibles líneas de investigación.

5.1 Conclusiones

La inteligencia artificial es una herramienta esencial en la búsqueda de nuevas técnicas y procedimientos que acelera los tiempos de computación y facilita la ejecución de tareas que, sin estos instrumentos, requerirían un mayor tiempo de dedicación por parte del ser humano.

En particular, este TFG ha demostrado que las técnicas de aprendizaje supervisado utilizadas (*Random Forest* y *kNN*) constituyen un mecanismo altamente efectivo para la predicción del tipo de buque. El papel del ser humano radica en analizar y optimizar dichos algoritmos, definiendo parámetros eficientes y filtrando los datos para lograr resultados más realistas.

En referencia a los objetivos marcados en el apartado 1.2 se han conseguido los mencionados a continuación:

- Las celdas H3 de Uber es un tipo de indexación geoespacial que se utiliza en muchos campos, entre los que destaca el análisis de trayectorias de los barcos. Sin embargo, su aplicación en modelos de aprendizaje supervisado para la detección de buques no se había explorado anteriormente, a diferencia de los parámetros estáticos y cinemáticos, cuya utilización para la predicción de la clase de buque es un tema de interés en la comunidad científica [70]. En este TFG se han conseguido implantar de diferentes maneras: su implicación directa, que como se explicó en el apartado 4.3 (trabajando con *kNN*) se descartaba por periodos de ejecución muy largos y su implicación mediante parámetros estadísticos basados en la frecuencia de aparición de las celdas y las distancias recorridas.
- Los resultados obtenidos en los experimentos con datos estáticos demuestran que son los atributos óptimos para la predicción del tipo de buque. Si bien el filtrado de parámetros repetidos ha hecho que los resultados obtenidos no sean en principio tan buenos como los expuestos en [4], el definir el modelo eliminando la técnica de *oversampling* y filtrando las muestras anómalas ha resultado clave para conseguir fiabilidad en el sistema propuesto.
- De igual forma, se han estudiado y analizado los datos dinámicos, y su aplicación en el modelo de predicción ha constituido la parte principal de este trabajo. Cabe destacar que, en general, los parámetros dinámicos son menos eficientes que los estáticos en las predicciones. Pero se ha demostrado que su aportación, en particular la de las celdas H3, y más en concreto, los atributos que se han definido como óptimos en el apartado 4.4, pueden llegar a ser fundamentales para definir el comportamiento de dos tipos de buque: remolcadores y pasajeros. Como se puede leer en el apartado 4.3, realizando experimentos solo con datos

cinemáticos, el modelo no era capaz de predecir con buenos resultados las clases de pasajeros, remolcadores y petroleros. Si bien, una vez se han añadido los atributos basados en las celdas H3, se consigue mejorar exponencialmente las dos primeras clases mencionadas, más en particular los barcos tipo remolcador. Esto es debido a los comportamientos tan singulares de estos tipos de barco, cuyo fin es trabajar en un área muy limitada generalmente cerca de puerto.

- En relación con los experimentos conjuntos se puede ver como los datos estáticos son los que marcan la diferencia en la predicción. Sin embargo, los cinemáticos ayudan a mejorar y terminar de definir los comportamientos de las clases de barcos mencionadas anteriormente.

En conclusión, se puede afirmar que la forma de los barcos varía dependiendo del tipo de buque, lo cual hace que la predicción utilizando únicamente estos parámetros sea muy efectiva. Además, es importante destacar que los algoritmos de aprendizaje supervisado son capaces de manejar un gran número de variables en tiempos razonables, lo que hace posible la incorporación de parámetros dinámicos en la predicción sin disminuir significativamente los tiempos de ejecución y, de esta manera, lograr una mayor precisión en los resultados obtenidos. En definitiva, la combinación de parámetros estáticos y dinámicos en los modelos de predicción puede resultar clave en la mejora de la eficiencia y eficacia de la clasificación de los buques. Por todo ello se constata que el empleo de las celdas H3 y los atributos definidos suponen una mejora de los resultados, más notable en los tipos de buque señalados.

5.2 Líneas futuras

Una vez definido el modelo óptimo para la predicción de tipo de buque en base a las áreas de actividad y datos AIS, se proponen las siguientes líneas futuras que permiten seguir trabajando en la línea en la que se incluye este trabajo:

- Creación de una base de datos de mayor tamaño capaz de alimentar el algoritmo para mejorar la calidad de predicción. Como se ha podido demostrar, la depuración de los datos tiene como consecuencia una disminución en el número de muestras. Además, el aumento de número de muestras ayuda favorablemente a la predicción del tipo de buque. Es por ello, por lo que se propone crear una base de datos que contenga información de diferentes épocas con el fin de conseguir definir de la forma más exacta los comportamientos de los diferentes tipos de buque.
- Validación del modelo e incorporación dentro del demostrador realizado por el CUD-ENM [2]. Como se mencionó en el capítulo 1.1, más de un tercio de los datos AIS no cubren dicho campo. Se propone aplicar el modelo y analizar su funcionamiento en tiempo real.
- Desarrollo de un sistema de información de clase de buque que se mantenga actualizado (en parte, con sistemas como el propuesto en este TFG) y al que puedan realizar consultas a tiempo real cualquier buque/unidad de la Armada.

En definitiva, las líneas de investigación que se pueden seguir son muy variadas. El conocimiento del entorno marítimo es algo fundamental para tener el poder naval y la inteligencia artificial es la herramienta clave para su desarrollo.

6 BIBLIOGRAFÍA

[1] Almirante General Jefe de Estado Mayor de la Armada , «Líneas Generales de la Armada 2022.,» [En línea].

[2] Miguel Rodelgo Lacruz, Belén Barragáns Martínez, Norberto Fernández García, George Higgins, Pablo Sendín Raña, Andrés Suárez Garcia, «Análisis de datos AIS en tiempo real para la detección de anomalías en el entorno marítimo,» CUD-ENM, 2021.

[3] «MarineTraffic,» [En línea]. Available: https://www.marinetraffic.com/es/ais/home/centerx:7.8/centery:38.3/zoom:5. [Último acceso: 14 marzo 2023].

[4] G. Rodríguez Casajús, «Predicción de tipo de buque utilizando datos AIS y técnicas de inteligencia artificial (Trabajo Fin de Grado),» 2022. [En línea]. Available: http://calderon.cud.uvigo.es/handle/123456789/530. [Último acceso: 19 enero 2023].

[5] Uber H3, «Overview of the H3 Geospatial Indexing System,» [En línea]. Available: https://h3geo.org/docs/core-library/overview/. [Último acceso: 24 enero 2023].

[6] Real Academia Española, Diccionario de la Lengua Española, Madrid: Espasa Calpe, 2000.

[7] P. N. Stuart Russell, Inteligencia Artificial, Un Enfoque Moderno, Mexico: Pearson Prentice Hall, 2008.

[8] R. González, «El test de Turing: dos mitos, un dogma.,» Scientific Electronic Library Online, 2007. [En línea]. Available: https://www.scielo.cl/scielo.php?pid=S0718-43602007000100003&script=sci_arttext. [Último acceso: 13 enero 2023].

[9] «Naukas, Turing y la inteligencia de lo no computable,» agosto 2019. [En línea]. Available: https://naukas.com/2019/08/27/turing-y-la-inteligencia-de-lo-no-computable/. [Último acceso: 12 enero 2023].

[10] National Geographic España, «Breve historia visual de la inteligencia artificial,» 2 diciembre 2020. [En línea]. Available: https://www.nationalgeographic.com.es/ciencia/breve-historia-visual-inteligencia-artificial_14419. [Último acceso: enero 2023].

[11] A. Fernández Candial, «Deep Blue-Kaspárov: cuando la máquina venció al hombre,» *La Vanguardia,* 10 febrero 2021.

[12] A. R. Aguiar, «Business Insider,» 22 marzo 2021. [En línea]. Available: https://www.businessinsider.es/debate-etica-ia-urgente-nunca-830479. [Último acceso: enero 2023].

[13] OpenAI, «ChatGPT,» [En línea]. Available: https://chat.openai.com/chat. [Último acceso: 13 febrero 2023].

[14] «Siri,» [En línea]. Available: https://www.apple.com/es/siri/. [Último acceso: 15 enero 2023].

[15] A. L. Samuel, «Some Studies in Machine Learning Using the Game of Checkers,» *IBM Journal of Research and Development,* vol. 3, pp. 210-229, 1959.

[16] Datascience Berkeley, «What is machine learning,» UC Berkeley School of Information, 26 junio 2020. [En línea]. Available: https://ischoolonline.berkeley.edu/blog/what-is-machine-learning/. [Último acceso: 16 enero 2023].

[17] H. Cenan, «What's the deal with AI, anyway?,» 12 julio 2019. [En línea]. Available: https://medium.com/zasti/whats-the-deal-with-ai-anyway-56a30177f438. [Último acceso: febrero 12 2023].

[18] L. Fridman, *Conference Deep Learning Basics: Introduction and Overview,* 2019.

[19] TIBCO, «¿Qué es el aprendizaje supervisado?,» [En línea]. Available: https://www.tibco.com/es/reference-center/what-is-supervised-learning#:~:text=El%20aprendizaje%20supervisado%20es%20una%20rama%20de%20Machine,tener%20que%20programar%20de%20manera%20expl%C3%ADcita%20d%C3%B3nde%20buscar.. [Último acceso: enero 2023].

[20] D. Calvo, «Aprendizaje no supervisado,» 19 marzo 2019. [En línea]. Available: https://www.diegocalvo.es/aprendizaje-no-supervisado/. [Último acceso: 17 enero 2023].

[21] Cleverdata, «Clustering, análisis de segmentos de clientes,» [En línea]. Available: https://cleverdata.io/clustering-analisis-de-segmentos-de-clientes-caso-de-exito-para-mazda/. [Último acceso: 15 enero 2023].

[22] Softtek, «La Reducción de Dimensionalidad en el Machine Learning,» 2 septiembre 2021. [En línea]. Available: https://softtek.eu/tech-magazine/artificial-intelligence/la-reduccion-de-dimensionalidad-en-el-machine-learning/. [Último acceso: 16 enero 2023].

[23] A. Ibáñez Martín, «Semi-Supervised Learning…el gran desconocido,» Telefónica Tech, 16 abril 2019. [En línea]. Available: https://empresas.blogthinkbig.com/semi-supervised-learningel-gran-desconocido/. [Último acceso: 18 enero 2023].

[24] S. Bhatt, «5 Things You Need to Know about Reinforcement Learning,» 28 Marzo 2018. [En línea]. Available: https://www.kdnuggets.com/2018/03/5-things-reinforcement-learning.html. [Último acceso: 19 enero 2023].

[25] E. Anguiano Batanero, «Aprendizaje por refuerzo y técnicas profundas (Trabajo Fin de Máster),» 2019. [En línea]. Available: https://repositorio.uam.es/bitstream/handle/10486/688799/anguiano_batanero_eloy_tfm.pdf?sequence=1. [Último acceso: enero 2023].

[26] E. Villarrubia, «Aprendizaje por refuerzo: área menos conocida del machine learning,» 13 junio 2022. [En línea]. Available: https://esi.uclm.es/index.php/2022/06/13/aprendizaje-por-refuerzo-area-menos-conocida-del-machine-learning/. [Último acceso: 18 enero 2023].

[27] IBM Watson Studio, «Supervised learning algorithms,» [En línea]. Available: https://www.ibm.com/topics/supervised-learning.

[28] IBM, «¿Qué es un árbol de decisión?,» [En línea]. Available: https://www.ibm.com/es-es/topics/decision-trees. [Último acceso: 30 enero 2023].

[29] V. Román, «Algoritmos Naive Bayes: Fundamentos e Implementación (Medium.com),» 25 abril 2019. [En línea]. Available: https://medium.com/datos-y-ciencia/algoritmos-naive-bayes-fudamentos-e-implementaci%C3%B3n-4bcb24b307f. [Último acceso: 31 enero 2023].

[30] V. Román, «Machine Learning Supervisado: Fundamentos de la Regresión Lineal,» 27 febrero 2019. [En línea]. Available: https://medium.com/datos-y-ciencia/machine-learning-supervisado-fundamentos-de-la-regresi%C3%B3n-lineal-bbcb07fe7fd. [Último acceso: 3 febrero 2023].

[31] A. Torres, «Aprendizaje Supervisado y Regresión Logística.,» 17 noviembre 2021. [En línea]. Available: https://www.freecodecamp.org/espanol/news/introduccion-al-aprendizaje-automatico/. [Último acceso: 2023 febrero 4].

[32] L. González, «Aprendizaje Supervisado: Support Vector Machine,» 23 marzo 2018. [En línea]. Available: https://aprendeia.com/aprendizaje-supervisado-support-vector-machine/. [Último acceso: 3 febrero 2023].

[33] Datagy, «Support Vector Machines (SVM) in Python with Sklearn,» 25 febrero 2022. [En línea]. Available: https://datagy.io/python-support-vector-machines/. [Último acceso: 3 febrero 2023].

[34] S. Raschka, «STAT 479: Machine Learning,» Department of Statistics, University of Wisconsin–Madison, Wisconsin, 2018.

[35] IBM, «¿Qué es el algoritmo de k vecinos más cercanos?,» [En línea]. Available: https://www.ibm.com/co-es/topics/knn. [Último acceso: 3 febrero 2023].

[36] O. Mbaabu, «Introduction to Random Forest in Machine Learning,» Section, 11 diciembre 2020. [En línea]. Available: https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/. [Último acceso: 5 febrero 2023].

[37] D. B. Itamar Reis, «Probabilistic Random Forest: A Machine Learning Algorithm for Noisy Data Sets,» *The Astronomical Journal*, p. 12, 2018.

[38] AprendeIA, «Métdo de agrupamiento o clustering,» 2023. [En línea]. Available: https://aprendeia.com/metodo-de-agrupamiento-o-clustering-aprendizaje-no-supervisado/#:~:text=El%20Clustering%20o%20agrupamiento%20es%20una%20de%20las,que%20no%20son%20aparentes%20para%20el%20ojo%20humanos.. [Último acceso: 5 febrero 2023].

[39] F. Sancho Caparrini, «Algoritmos de Clustering,» 20 diciembre 2020. [En línea]. Available: https://www.cs.us.es/~fsancho/?e=230. [Último acceso: 6 febrero 2023].

[40] Ciencia de datos, «Hierarchical Clustering,» [En línea]. Available: https://www.cienciadedatos.net/documentos/py20-clustering-con-python.html. [Último acceso: 6 febrero 2023].

[41] F. Berzal, «Clustering jerárquico, Universidad de Granada,» [En línea]. Available: https://elvex.ugr.es/idbis/dm/slides/42%20Clustering%20-%20Hierarchical.pdf. [Último acceso: 6 febrero 2023].

[42] Universidad de Oviedo, «El algoritmo k-means aplicado a clasificación y procesamiento de imágenes,» [En línea]. Available: https://www.unioviedo.es/compnum/laboratorios_py/kmeans/kmeans.html. [Último acceso: 6 febrero 2023].

[43] DataScientest, «Machine Learning & Clustering: el algoritmo DBSCAN,» 30 noviembre 2022. [En línea]. Available: https://datascientest.com/es/machine-learning-clustering-dbscan. [Último acceso: 6 febrero 2023].

[44] V. Manneni, «Clustering algorithms for segmentation,» [En línea]. Available: http://statsvenu.com/clustering-algorithms-for-segmentation/. [Último acceso: 6 febrero 2023].

[45] Z. Jaadi, «A Step-by-Step Explanation of Principal Component Analysis (PCA),» 26 septiembre 2022. [En línea]. Available: https://builtin.com/data-science/step-step-explanation-principal-component-analysis. [Último acceso: 7 febrero 2023].

[46] S. Rawat, «Introduction to Independent Component Analysis in Machine Learning,» 10 septiembre 2021. [En línea]. Available: https://www.analyticssteps.com/blogs/introduction-independent-component-analysis-machine-learning. [Último acceso: 7 febrero 2023].

[47] Armada , «Conocimiento del Entorno Marítimo Fundamento,» [En línea]. Available: https://1library.co/article/conocimiento-del-entorno-mar%C3%ADtimo-fundamento.zxx56n4z. [Último acceso: 10 febrero 2023].

[48] A. Catao, «Plataforma continental,» Pinterest, [En línea]. Available: https://www.pinterest.com/pin/554083560394232478/. [Último acceso: 11 febrero 2023].

[49] M. de la Gándara García, «El COVAM de la Armada al servicio de la comunidad,» Instituto Español de Estudios Estratégicos, 27 julio 2011. [En línea]. Available: https://www.ieee.es/Galerias/fichero/docs_opinion/2011/DIEEEO55-2011COVAM_DeArmada.pdf. [Último acceso: 11 febrero 2023].

[50] J. Máiz, «Centro de operaciones y vigilancia de acción marítima de la Armada,» *Revista Defensa nº 420,* abril 2013.

[51] ENCOMAR, «Cooperación seguridad marítima,» Armada, [En línea]. Available: https://encomar.covam.es/registro-de-transito. [Último acceso: 11 febrero 2023].

[52] C. Grima, «El diagrama de Voronoi, la forma matemática de dividir el mundo,» 8 mayo 2017. [En línea]. Available: https://www.abc.es/ciencia/abci-diagrama-voronoi-forma-matematica-dividir-mundo-201704241101_noticia.html. [Último acceso: 21 ENERO 2023].

[53] «Uber H3 para análisis de datos con Python,» [En línea]. Available: https://ichi.pro/es/uber-h3-para-analisis-de-datos-con-python-205876586619166#:~:text=H3%20es%20un%20marco%20de%20c%C3%B3digo%20abier

to%20desarrollado,obtener%20conocimientos%20de%20grandes%20conjuntos%20de%20datos%20geoespaciales.. [Último acceso: 21 enero 2023].

[54] E. Vankat, «Towards Data Science, Uber H3 for Data Analysis with Python,» 11 enero 2021. [En línea]. Available: https://towardsdatascience.com/uber-h3-for-data-analysis-with-python-1e54acdcc908. [Último acceso: 23 enero 2023].

[55] Uber H3, «Indexing,» [En línea]. Available: https://h3geo.org/docs/3.x/highlights/indexing. [Último acceso: 24 enero 2023].

[56] Uber H3, «S2 vs H3 Comparison,» [En línea]. Available: https://h3geo.org/docs/3.x/comparisons/s2. [Último acceso: 24 enero 2023].

[57] Uber H3, «ZIP Codes vs H3 comparison,» [En línea]. Available: https://h3geo.org/docs/3.x/comparisons/admin. [Último acceso: 24 enero 2023].

[58] V. Alonso Aller, «Análisis de los sistemas de indexado geoespacial para el conocimiento del entorno marítimo,» 30 julio 2021. [En línea]. Available: http://calderon.cud.uvigo.es/xmlui/handle/123456789/402?show=full. [Último acceso: 12 febrero 2023].

[59] Jianxin Wang, Jian Chen, Jianmin Liu, «A Ship Type Classification Method Based on AIS and SAR Information",» *Journal of Marine Science and Application,* 2015.

[60] Ioanna N. Liritzis, Georgios A. Papageorgiou, «Marine Traffic Prediction using Machine Learning Algorithms and AIS Data,» *Journal of Marine Systems,* 2019.

[61] «Jupyter Lab,» [En línea]. Available: https://jupyter.org. [Último acceso: 15 03 2023].

[62] Digital Guide IONOS, «Jupyter Notebook: documentos web para análisis de datos, código en vivo y mucho más,» 28 febrero 2019. [En línea]. Available: https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/jupyter-notebook/. [Último acceso: 2023 febrero 19].

[63] «Python,» [En línea]. Available: https://www.python.org/. [Último acceso: 10 03 2023].

[64] «SQLite3,» [En línea]. Available: https://www.sqlite.org/index.html. [Último acceso: 13 03 2023].

[65] SQLite, «What Is SQLite?,» [En línea]. Available: https://www.sqlite.org/index.html. [Último acceso: 21 febrero 2023].

[66] MarineTraffic, «What kind of information is AIS-transmitted?,» [En línea]. Available: https://help.marinetraffic.com/hc/en-us/articles/205426887-What-kind-of-information-is-AIS-transmitted-. [Último acceso: 12 febrero 2023].

[67] Navigation Centre US Coast Guard, «AIS Messages,» [En línea]. Available: https://www.navcen.uscg.gov/ais-messages. [Último acceso: 19 febrero 2023].

[68] Zhenguo Yan, Xin Song , Hanyang Zhong, Lei Yang and Yitao Wang, «Ship Classification and Anomaly Detection Based on Spaceborne AIS Data Considering Behavior Characteristics,» Sensors, 11 octubre 2022. [En línea]. Available: https://www.mdpi.com/1424-8220/22/20/7713. [Último acceso: 25 febrero 2023].

[69] Paul Kraus, Camilla Mohrdieck, Friedhelm Schwenker, «Ship classification based on trajectory data with machine-learning methods,» Airbus Defense and Space, Bonn, Germany, 2018.

[70] Yitao Wang, Lei Yang, Xin Song and Xuan Li, «Ship classification based on random forest using static information from AIS data,» College of Aerospace Science and Engineering , Hunan, China, 2021.

[71] F. Tseroni, «Vessel details that can be changed in AIS transponders,» MarineTraffic Blog, 6 mayo 2019. [En línea]. Available: https://www.marinetraffic.com/blog/vessel-details-that-can-be-changed-in-ais-transponders/. [Último acceso: 20 marzo 2022].

[72] J. M. Heras, «Precision, Recall, F1, Accuracy en clasificación,» IArtificial.net, 9 octubre 2020. [En línea]. Available: https://www.iartificial.net/precision-recall-f1-accuracy-en-clasificacion/. [Último acceso: 3 marzo 2023].

ANEXO I: IMPLICACIONES SOCIALES, Y/O ECONÓMICAS, Y/O AMBIENTALES

El desarrollo de este Trabajo de Fin de Grado se ha basado en el estudio y análisis de diferentes parámetros en la predicción de tipo de buque. En particular en la influencia de las celdas H3 en la predicción mencionada anteriormente. Por ello en este caso no procede su valoración en este trabajo, ya que no se han percibido ninguna influencia a nivel ambiental/económico.

En cuanto a implicaciones sociales, dicho trabajo nace, como se explica en el capítulo 1, de un proyecto realizado por el COVAM y el CUD-ENM cuyo fin último es la detección de anomalías en el comportamiento de los barcos. En este ámbito se podría llegar a plantear cierta influencia, ya que la predicción de tipo de buque puede ayudar a la identificación de dichas anomalías, sobre las cuales el COVAM podría llegar a tomar algún tipo de acción. Se podría valorar que este TFG pudiese contribuir en la mejora de los procedimientos de vigilancia del COVAM, suponiendo un ejemplo de contribución a la sociedad.

ANEXO II: REFLEXIONES ÉTICAS Y SOCIALES

La inteligencia artificial es una tecnología en constante evolución que tiene un gran potencial para transformar la sociedad. Sin embargo, su creciente uso también plantea preocupaciones éticas y morales, que deben ser consideradas y abordadas para garantizar que la IA sea utilizada de manera responsable y beneficiosa para todos.

Uno de los principales problemas éticos asociados con la IA es la falta de transparencia en la toma de decisiones automatizadas. La IA se basa en algoritmos complejos que pueden tomar decisiones sin la intervención humana directa. Por ejemplo, los coches autónomos, donde se plantea el dilema de si un coche tuviese que elegir entre atropellar a un anciano o a un niño en caso de accidente. Pues bien, la respuesta no es tan sencilla, muchos podrían pensar que lo óptimo sería atropellar al anciano ya que es quien ha vivido más y en términos de justicia el niño le queda mucho por vivir. Sin embargo, en otra cultura como puede ser la asiática, los ancianos son muy venerados y en su caso probablemente lo óptimo sería atropellar al niño. Adicionalmente, una vez que el coche ha atropellado a una persona por un fallo en el sistema de IA, se plantea el segundo dilema, la responsabilidad. Si la IA toma decisiones importantes que afectan a la vida de las personas, ¿quién es responsable si se toma una decisión incorrecta? ¿Es el creador del algoritmo, el propietario del sistema, el usuario final o la IA misma?

Además, la IA puede perpetuar y ampliar la discriminación existente en la sociedad. Si los algoritmos se entrenan con datos sesgados, pueden perpetuar prejuicios y discriminación en la toma de decisiones automatizadas. Por ejemplo, los algoritmos utilizados en la selección de candidatos pueden basarse en datos históricos que reflejen la discriminación en la contratación, lo que puede perpetuar esa discriminación en el futuro.

Otro problema ético relacionado con la IA es la privacidad. La IA puede procesar grandes cantidades de datos personales, lo que plantea preocupaciones sobre cómo se manejan y protegen estos datos. Si la IA se utiliza para tomar decisiones importantes sobre las personas, como las decisiones de crédito o empleo, entonces la privacidad se convierte en un asunto crítico.

Por último, la IA también plantea preocupaciones éticas en términos de empleo. La IA puede automatizar trabajos que antes eran realizados por seres humanos, lo que puede tener un impacto significativo en el empleo y la calidad de vida de las personas. Si bien la automatización puede liberar a las personas de tareas repetitivas y peligrosas, también puede resultar en una pérdida neta de empleo y una mayor desigualdad económica.

En conclusión, la IA es una tecnología en constante evolución que plantea una serie de problemas éticos y morales que deben ser abordados para garantizar que se utilice de manera responsable y beneficiosa para todos. La transparencia, la discriminación, la responsabilidad y el empleo son solo algunos de los problemas clave que deben ser abordados para garantizar que la IA se utilice de manera ética y responsable en el futuro.

En cualquier caso, en el trabajo desarrollado aquí se contribuye a mejorar una línea de investigación que persigue como fin último proporcionar información a expertos sobre determinados comportamientos anómalos, pero son los operadores humanos los que tomarían las decisiones de actuaciones futuras.

ANEXO III: DICCIONARIO DE SIGLAS, ACRÓNIMOS Y ABREVIATURAS

-AJEMA: Almirante General Jefe de Estado Mayor de la Armada.

-CUD: Centro Universitario de la Defensa.

-ENM: Escuela Naval Militar.

-AIS: *Automatic Identification System* (Sistema de Identificación Automática).

-COVAM: Centro de Operaciones y Vigilancia de Acción Marítima.

-IA: Inteligencia Artificial.

-CEMAI: Inteligencia Artificial para el Conocimiento del Entorno Marítimo.

-SIRENA: Sistema de Inteligencia artificial para el Reconocimiento del ENtorno mArítimo.

-RAE: Real Academia Española.

-ML: *Machine Learning* (Aprendizaje Automático).

-DL: *Deep Learning* (Aprendizaje Profundo).

-MIT: *Massachusetts Institute of Technology* (Instituto de Tecnología de Massachusetts).

-PCA: *Principal Component Analysis* (Análisis de Componentes Principales).

-ICA: *Independent Component Analysis* (Análisis de Componentes Independientes).

-SVM: *Support Vector Machine* (Máquinas de Vector Soporte).

-*kernel:* intérprete de líneas de *Pyhton.*

-kNN: *k-Nearest Neighbor* (*k* vecinos más cercanos).

-RF: *Random Forest* (Bosque Aleatorio).

-*cluster:* grupo.

-DBSCAN: *Desnsity-Based Spatial Clustering of Applications with Noise* (Agrupación espacial basada en la densidad de aplicaciones con ruido).

-CEM: Conocimiento Entorno Marítimo.

-ZEE: Zona Económica Exclusiva.

-FAM: Fuerza de Acción Marítima.

-ALMART: Almirante de Acción Marítima.

-CMOM: Comandante del Mando Operativo Marítimo.

-MSA: *Maritime Situational Awareness* (Conocimiento de la situación marítima)

-*RMP: Recognized Maritime Picture.*

-AN: Alférez de Navío.

-MMSI: *Maritime Mobile Service Identity* (número de identificación del servicio móvil marítimo).

-SQL: *Structured Query Language* (Lenguaje de consulta estructurado).

-SOG: *Speed Over Groun*d (velocidad sobre el fondo).

-COG: *Course Over Ground* (rumbo sobre fondo).

-IMO: *International Maritime Organization* (Organización Marítima Internacional (OMI))

ANEXO IV: CÓDIGO NUMÉRICO EN FUNCIÓN TIPO DE BUQUE

CÓDIGO	TIPO DE BUQUE
20-29	*Win in ground (WIG)*
30	*Fishing*
31	*Towing*
32	*Towing: length exceeds 200 m or breadth exceeds 25m*
33	*Dredging or underwater ops.*
34	*Diving ops.*
35	*Military ops.*
36	*Sailing*
37	*Pleasure Craft*
40-49	*High speed craft (HSC)*
50	*Pilot Vessel*
51	*Search and Rescue vessel*
52	*Tug*
53	*Port Tender*
54	*Anti-pollution equipment*
55	*Law Enforcement*
56	*Spare – Local Vessel*
57	*Spare – Local Vessel*
58	*Medical Transport*
59	*Noncombatant ship according to RR Resolution No. 18.*
60-69	*Passenger*

Tabla IV-1. Código tipo de buque datos AIS

ANEXO V: CÓDIGO *DATOS_ESTATICOS.IPYNB*

```
    #Librerias
    import sqlite3
    import numpy as np
    from tqdm import tqdm
    import seaborn as sns
    import matplotlib.pyplot as plt

    #Funciones predeterminadas
    from sklearn.metrics import confusion_matrix
    from sklearn.model_selection import train_test_split, GridSearchCV
    from sklearn.preprocessing import MinMaxScaler
    from sklearn.metrics import classification_report
    from sklearn.ensemble import RandomForestClassifier
    from sklearn.pipeline import Pipeline
    from imblearn.over_sampling import SMOTE
    from imblearn.over_sampling import RandomOverSampler
    from imblearn.under_sampling import RandomUnderSampler

    #Funcion atributos estáticos
    def statics(df):
        df['len']=df.a+df.b
        df['wid']=df.c+df.d
        df['ldivw']=df.len/df.wid
        df['ldivd']=df.len/df.draught
        df['wdivd']=df.wid/df.draught
        df['area']= df.len*df.wid
        df['grith']= df.len+df.wid
        df['aml']=df.len*df.draught
        df['amt']=df.wid*df.draught
        df['vs']=df.len*df.draught*df.wid
        df['aol']=df.a/df.len
        return df

    #Base de datos
    path_data ='../FINAL_STATIC/datos_estaticos.sql'
    por = sqlite3.connect(path_data)
    cur = por.cursor()

    #Lectura CSV y creación atributos estáticos
    cols=['MMSI', 'to_bow', 'to_stern', 'to_port', 'to_starboard', 'draught',
'shiptype']
    ncol=['mmsi', 'a', 'b', 'c', 'd', 'draught', 'shiptype']
    data=pd.read_csv('../ais_static_20230116.csv', sep=',', chunksize=200000)
    for df in data:
        chunk=df[cols]
        chunk.columns=ncol
        atributos=statics(chunk)
    atributos=atributos[atributos.replace([np.inf,-np.inf],
np.nan).notnull().all(axis=1)]
    atributos=atributos.dropna()
    df=atributos[atributos.shiptype.isin(['MERCANTE','PETROLERO','PESQUERO','PASA
JEROS','REMOLCADOR'])]

    #Eliminan datos repetidos y inchorentes
```

```
df = df[df.replace([np.inf, -np.inf], np.nan).notnull().all(axis=1)]
df = df.dropna()
df = df.drop_duplicates()
df = df[df['draught'] != 0]
df = df[df['mmsi'] != 0]
df = df[(df['a'] != 0) & (df['b'] != 0)]
df = df[(df['c'] != 0) & (df['d'] != 0)]
df.to_sql('static_data', con=por, if_exists='append', index=False)

#Se lee base de datos varios días y se eliminan outliers
df = pd.read_sql("""SELECT * FROM static_data""", con=por)
df = df[df.replace([np.inf, -np.inf], np.nan).notnull().all(axis=1)]
df = df.dropna()
df = df.drop_duplicates()
ship_types =['MERCANTE','PETROLERO','PESQUERO','PASAJEROS','REMOLCADOR']
atributos = ['a','b','len','ldivw','area','grith','aol']
for m in ship_types:
    for n in atributos:
        q = df[df['shiptype'] == m][n].quantile(0.98)
        df[(df['shiptype'] == m) & (df[n] > q)] = None
df = df.dropna()

#Division entrtanamiento y test en base al mmsi
dfa = df.filter(items=['mmsi'])
dfa = dfa. drop_duplicates()
df_tr, df_ts = train_test_split(dfa, random_state = 42)
df_train = pd.merge(df_tr, df)
df_train = df_train.drop('mmsi', axis=1)
df_test = pd.merge(df_ts, df)
df_test = df_test.drop('mmsi', axis=1)

#División parámetros dependientes y dependientes
X_train = df_train.drop('shiptype', axis=1)
X_test = df_test.drop('shiptype', axis=1)
y_train = df_train.shiptype
y_test = df_test.shiptype

#Se crea el modelo
pipe=Pipeline([('scaler',MinMaxScaler()),('classifier',RandomForestClassifier
(n_estimators=100))])
model=pipe.fit(X_train, y_train)
y_pred=model.predict(X_test)

#Resultados
print(classification_report(y_test, y_pred, zero_division=0))

forest=model['classifier']
importances=forest.feature_importances_
std=np.std([tree.feature_importances_ for tree in forest.estimators_], axis=0)
fi=pd.Series(importances, index=X_train.columns).reset_index()
fi.columns=['att', 'importance']
fi=fi.sort_values(by='importance', ascending=False).reset_index(drop=True)
g=sns.barplot(data=fi, x='att', y='importance', color='g')
g.figure.set_size_inches(7,4)

cur.close()
por.close()
```

ANEXO VI: CÓDIGO *DATOS_DINAMICOS.IPYNB*

```
#Librerias
import pandas as pd
import sqlite3
import numpy as np
import seaborn as sns
from tqdm import tqdm
import matplotlib.pyplot as plt
import folium
import sqlalchemy
import h3

#Funciones predeterminadas
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import classification_report
from sklearn.ensemble import RandomForestClassifier
from sklearn.pipeline import Pipeline
from imblearn.over_sampling import SMOTE
from imblearn.over_sampling import RandomOverSampler
from imblearn.under_sampling import RandomUnderSampler

#Base de datos
path_data = r'../FINAl_DINAMIC/datos_dinamicos.sql'
conn = sqlite3.connect(path_data)
cur = conn.cursor()

#Definicion de funciones

def cinematics(df):
    #tiempo
    df = df.sort_values(by='tim')
    df.tim = pd.to_datetime(df.tim,unit='ms')
    df['dtime'] = df.tim.diff().dt.seconds
    hours = df.dtime/3600
    #lineal
    dy = np.radians(df.lat.diff())*6373/1.852
    dx = np.radians(df.lon.diff())*6373/1.852
    df['nmi'] = np.sqrt(dx**2 + dy**2)
    df['sog_hat'] = df.nmi/hours
    df['sog_err'] = (df.sog_hat-df.sog)/(df.sog+0.1)*100
    df['acc_hat'] = df.sog_hat.diff()/hours
    df['jerk_hat'] = df.acc_hat.diff()/hours
    df['acc'] = df.sog.diff()/hours
    df['jerk'] = df.acc.diff()/hours
    #angular
    deg_norm = df.cog.diff()%360
    deg_diff = pd.concat([360-deg_norm, deg_norm], axis=1).min(axis=1)
    df['omega'] = np.radians(deg_diff)/hours
    df['alpha'] = df.omega.diff()/hours
    df['zeta'] = df.alpha.diff()/hours
    return df
```

```
def aggregate(df):
    #aceleración
    df['avg_acc'] = df['acc'].mean()
    df['sdv_acc'] = df['acc'].std()
    df['iod_acc'] = df['acc'].var()/ df['avg_acc']
    df['q50_acc'] = df['acc'].quantile(0.50)
    df['skw_acc'] = df['acc'].skew()
    df['krt_acc'] = df['acc'].kurt()
    #sog
    df['avg_sog'] = df['sog'].mean()
    df['sdv_sog'] = df['sog'].std()
    df['iod_sog'] = df['sog'].var()/ df['avg_sog']
    df['q50_sog'] = df['sog'].quantile(0.50)
    df['skw_sog'] = df['sog'].skew()
    df['krt_sog'] = df['sog'].kurt()
    #sog_hat
    df['avg_sog_hat'] = df['sog_hat'].mean()
    df['sdv_sog_hat'] = df['sog_hat'].std()
    df['iod_sog_hat'] = df['sog_hat'].var()/ df['avg_sog_hat']
    df['q50_sog_hat'] = df['sog_hat'].quantile(0.50)
    df['skw_sog_hat'] = df['sog_hat'].skew()
    df['krt_sog_hat'] = df['sog_hat'].kurt()
    #sog_err
    df['avg_sog_err'] = df['sog_err'].mean()
    df['sdv_sog_err'] = df['sog_err'].std()
    df['iod_sog_err'] = df['sog_err'].var()/df['avg_sog_err']
    df['q50_sog_err'] = df['sog_err'].quantile(0.50)
    df['skw_sog_err'] = df['sog_err'].skew()
    df['krt_sog_err'] = df['sog_err'].kurt()
    #acc_hat
    df['avg_acc_hat'] = df['acc_hat'].mean()
    df['sdv_acc_hat'] = df['acc_hat'].std()
    df['iod_acc_hat'] = df['acc_hat'].var()/ df['avg_acc_hat']
    df['q50_acc_hat'] = df['acc_hat'].quantile(0.50)
    df['skw_acc_hat'] = df['acc_hat'].skew()
    df['krt_acc_hat'] = df['acc_hat'].kurt()
    #jerk_hat
    df['avg_jerk_hat'] = df['jerk_hat'].mean()
    df['sdv_jerk_hat'] = df['jerk_hat'].std()
    df['iod_jerk_hat'] = df['jerk_hat'].var()/ df['avg_jerk_hat']
    df['q50_jerk_hat'] = df['jerk_hat'].quantile(0.50)
    df['skw_jerk_hat'] = df['jerk_hat'].skew()
    df['krt_jerk_hat'] = df['jerk_hat'].kurt()
    #jerk
    df['avg_jerk'] = df['jerk'].mean()
    df['sdv_jerk'] = df['jerk'].std()
    df['iod_jerk'] = df['jerk'].var()/df['avg_jerk']
    df['q50_jerk'] = df['jerk'].quantile(0.50)
    df['skw_jerk'] = df['jerk'].skew()
    df['krt_jerk'] = df['jerk'].kurt()
    #omega
    df['avg_omega'] = df['omega'].mean()
    df['sdv_omega'] = df['omega'].std()
    df['iod_omega'] = df['omega'].var()/  df['avg_omega']
    df['q50_omega'] = df['omega'].quantile(0.50)
    df['skw_omega'] = df['omega'].skew()
    df['krt_omega'] = df['omega'].kurt()
    #alpha
```

```
    df['avg_alpha'] = df['alpha'].mean()
    df['sdv_alpha'] = df['alpha'].std()
    df['iod_alpha'] = df['alpha'].var()/ df['avg_alpha']
    df['q50_alpha'] = df['alpha'].quantile(0.50)
    df['skw_alpha'] = df['alpha'].skew()
    df['krt_alpha'] = df['alpha'].kurt()
    #zeta
    df['avg_zeta'] = df['zeta'].mean()
    df['sdv_zeta'] = df['zeta'].std()
    df['iod_zeta'] = df['zeta'].var()/df['avg_zeta']
    df['q50_zeta'] = df['zeta'].quantile(0.50)
    df['skw_zeta'] = df['zeta'].skew()
    df['krt_zeta'] = df['zeta'].kurt()
    return df

def dinamic(df):
    df_dinamic = df.drop('acc', axis=1)
    df_dinamic = df_dinamic.drop('sog', axis=1)
    df_dinamic = df_dinamic.drop('sog_hat', axis=1)
    df_dinamic = df_dinamic.drop('sog_err', axis=1)
    df_dinamic = df_dinamic.drop('acc_hat', axis=1)
    df_dinamic = df_dinamic.drop('jerk_hat', axis=1)
    df_dinamic = df_dinamic.drop('jerk', axis=1)
    df_dinamic = df_dinamic.drop('omega', axis=1)
    df_dinamic = df_dinamic.drop('alpha', axis=1)
    df_dinamic = df_dinamic.drop('zeta', axis=1)
    return df_dinamic

def celdas(df_h3, dfu):


    df_h3['h3_7'] = df_h3['h3_9'].apply(lambda x: h3.h3_to_parent(x, 7))
    df_h3['h3_7'] = df_h3['h3_7'].apply(lambda x: int(x, 16))


    #Creo tabla frecuencias celdas nivel 9
    dfe1 = df_h3.groupby('imo')['octal_h3'].value_counts().reset_index(name =
'freq9')
    dfe1.to_sql('nivel9', con = conn, if_exists = 'replace', index = False)

    #Creo tabla frecuencias celdas nivel 7
    dfe2 = df_h3.groupby('imo')['h3_7'].value_counts().reset_index(name =
'freq7')
    dfe2.to_sql('nivel7', con = conn, if_exists = 'replace', index = False)

    #Calculo parámetros estadisticos de las celdasH3
    for m in tqdm(dfu):
        dft1 = pd.read_sql(f"""SELECT imo, freq9 FROM nivel9 WHERE imo={m}""",
con=conn)
        dft1 = statistics9(dft1)

        dft2 = pd.read_sql(f"""SELECT imo, freq7  FROM nivel7 WHERE imo={m}""",
con=conn)
        dft2 = statistics7(dft2)

        dft = pd.merge(dft1,dft2)
```

```
        dft.to_sql('estadisticos_celdas', con=conn, if_exists='append',
index=False)


    df_final = pd.read_sql(f"""SELECT * FROM estadisticos_celdas""", con=conn)
    df_final = df_final[df_final.replace([np.inf, -np.inf],
np.nan).notnull().all(axis=1)]
    df_final = df_final.dropna()
    df_final = df_final.drop_duplicates()

    return df_final

def statistics9(df):
    df['h3_meanfreq'] = df['freq'].mean()
    df['h3_stdfreq'] = df['freq'].std()
    df['h3_iod'] = df['freq'].var()/ df['h3_meanfreq']
    df['h3_q50freq'] = df['freq'].quantile(0.50)
    df['h3_skwfreq'] = df['freq'].skew()
    df['h3_krtfreq'] = df['freq'].kurt()
    return df

def statistics7(df):
    df['h3_meanfreq'] = df['freq'].mean()
    df['h3_stdfreq'] = df['freq'].std()
    df['h3_iod'] = df['freq'].var()/ df['h3_meanfreq']
    df['h3_q50freq'] = df['freq'].quantile(0.50)
    df['h3_skwfreq'] = df['freq'].skew()
    df['h3_krtfreq'] = df['freq'].kurt()
    return df

def distance_H3(df, dfu):

    df_dist = pd.DataFrame(columns=['imo', 'H3maxdistance', 'H3distanceFL',
'h3pathdistance'])
    ships = df.groupby('imo')['h3_9'].apply(list)
    for imo in tqdm(dfu):
        h3distance = 0
        celdas = ships[imo]
        trayectoria = trayectoriaH3(celdas)
        try:
            h3distanceFL = h3.h3_distance(celdas[0], celdas[len(celdas)-1])
        except:
            h3distanceFL = -1
        h3distance = max_distance(celdas, h3distance)

        df_new = {'imo': imo, 'H3distance': h3distance, 'H3distanceBL':
h3distanceFL, 'h3pathdistance': trayectoria }
        df_dist = pd.concat([df_dist, pd.DataFrame(df_new, index=[0])],
ignore_index=True)

    df_dist = df_dist[df_dist.replace([np.inf, -np.inf],
np.nan).notnull().all(axis=1)]
    df_dist = df_dist.dropna()
    df_dist = df_dist.drop_duplicates()
    df_dist['h3difdistance'] = abs(df_dist['H3distance']- df_dist['H3distance'])

    return df_dist
```

```
def max_distance(celdas, h3distance):
    for i in range(0, len(celdas)-1):
        for j in range(i+1, len(celdas)):
            try:
                dist = h3.h3_distance(celdas[i], celdas[j])
                if dist > h3distance:
                    h3distance = dist
            except:
                h3distance = -1
                return h3distance

    return h3distance

def trayectoriaH3(celdas):
    trayectoria=0
    for j in range(len(celdas)-1):
        try:
            trayectoria=trayectoria+h3.h3_distance(celdas[i],celdas[i+1])
        except:
            return -1;
    return trayectoria;

def celdas_merge(df_h3, df_celdas, df_distance):

    df = df_h3.filter(items=['imo', 'shiptype'])
    df = df.drop_duplicates()
    df_total = pd.merge(df_celdas, df_distance)
    df_total = pd.merge(df, df_total)

    return df_total

#Leer csv dinammicos

orig = ['TIMESTAMP', 'IMO', 'LATITUDE', 'LONGITUDE', 'H3_9', 'OCTAL_H3_9',
'COG', 'SOG', 'SHIPTYPE AIS']
subs = ['tim', 'imo', 'lat', 'lon', 'h3_9', 'octal_h3', 'cog', 'sog',
'shiptype']

reader = pd.read_csv('../ais20230201.csv', sep=',', chunksize=1000000)
for df in reader:
    chunk=df[orig]
    chunk.columns = subs
    chunk=chunk[chunk.replace([np.inf, -np.inf], np.nan).notnull().all(axis=1)]
    chunk=chunk.dropna()
    df =
chunk[chunk.shiptype.isin(['MERCANTE','PETROLERO','PESQUERO','PASAJEROS',
'REMOLCADOR'])]
    df=df[df['imo']!=0]
    df=df[df['imo']!='NULL']
    df=df[df['sog']>0]
    df=df[df['sog']<130]
    df.to_sql('basic_data', con=conn, if_exists='append', index=False)
#Estadisticos celdas H3
df_h3 = pd.read_sql('SELECT imo, h3_9, octal_h3, shiptype FROM basic_data',
con=conn)
df_h3 = df_h3[df_h3.replace([np.inf, -np.inf], np.nan).notnull().all(axis=1)]
df_h3 = df_h3.dropna()
dfu_h3 = df_h3['imo'].unique()
```

```
df_celdas = celdas(df_h3, dfu_h3)
df_distance = distance_H3(df_h3, dfu_h3)
df_total = celdas_merge(df_h3, df_celdas, df_distance)
df_total_H3.to_sql('celdasH3_data', con=conn, if_exists='replace', index=False)

#Estadisticos dinámicos
df = pd.read_sql('SELECT tim, imo, lat, lon, cog, sog, shiptype FROM
basic_data', con=conn)
df = df[df.replace([np.inf, -np.inf], np.nan).notnull().all(axis=1)]
df = df.dropna()
dfu = df_total_H3['imo'].unique()
for m in tqdm(dfu):
    dfc = pd.read_sql(f"""SELECT tim, imo, cog, sog, lat, lon, shiptype FROM
basic_data WHERE imo={m}""", con=conn)
    dfc= cinematics(dfc)

    dfc.to_sql('pre_dinamic', con=conn, if_exists='append', index=False)

df = pd.read_sql(f""" SELECT * FROM pre_dinamic""", con=conn)
df = df[df.replace([np.inf, -np.inf], np.nan).notnull().all(axis=1)]
df = df.dropna()

dfu = df['imo'].unique()

for m in tqdm(dfu):
    dfe = pd.read_sql(f"""SELECT imo, acc, sog, sog_hat, sog_err, acc_hat,
jerk_hat, jerk, omega, alpha, zeta, shiptype
    FROM pre_dinamic WHERE imo={m} AND sog<>0 AND sog_hat<>0 AND sog_hat NOT
NULL AND acc<>0 AND acc NOT NULL AND omega<>0
    AND omega NOT NULL AND jerk<>0 AND jerk NOT NULL AND sog_err<>0 AND sog_err
NOT NULL AND acc_hat<>0 AND acc_hat NOT NULL
    AND jerk_hat<>0 AND jerk_hat NOT NULL AND alpha<>0 AND alpha NOT NULL AND
zeta<>0 AND zeta NOT NULL""", con=conn)
    dfe = aggregate(dfe)
    dfe.to_sql('complete_dinamic', con=conn, if_exists='append', index=False)

df = pd.read_sql('SELECT * FROM complete_dinamiC', con=conn)
df = df[df.replace([np.inf, -np.inf], np.nan).notnull().all(axis=1)]
df = df.dropna()

df_total_dinamic = dinamic(df)
df_total_dinamic = df_total_dinamic.drop_duplicates()
df_total_dinamic.to_sql('dinamic_data', con=conn, if_exists='replace',
index=False)

#Unir datos dinámicos y celdas H3
df_total_H3 = pd.read_sql('SELECT * FROM celdasH3_data', con=conn)
df_total_H3

df_total_dinamic = pd.read_sql('SELECT * FROM dinamic_data', con=conn)
df_total_dinamic

df_TOTAL = pd.merge(df_total_dinamic, df_total_H3)
df_TOTAL = df_TOTAL.drop_duplicates()
df_TOTAL.to_sql('TOTAL_data', con=conn, if_exists='replace', index=False)

#Eliminan outliers
df = pd.read_sql('SELECT * FROM TOTAL_data', con=conn)
```

```
shiptypes = ['PETROLERO','MERCANTE','PESQUERO','REMOLCADOR', 'PASAJEROS']
atributos = df.select_dtypes(include=np.number).columns.tolist()



for m in shiptypes:
    for n in atributos:
        q = df[df['shiptype'] == m][n].quantile(0.98)
        df[(df['shiptype'] == m) & (df[n] > q)] = None
        df.loc[(df['shiptype'] == m) & (df[n] > q)] = None
df = df.dropna()
df.to_sql(DEPURATE_data', con=conn, if_exists='replace', index=False)

df = pd.read_sql('SELECT * FROM DEPURATE_data', con=conn)
df = df[df.replace([np.inf, -np.inf], np.nan).notnull().all(axis=1)]
df = df.dropna()

#División entrenamiento y test
df = df.drop('imo', axis=1)
df_train, df_test = train_test_split(df, random_state = 42)

X_train=df_train.drop('shiptype', axis=1)
y_train=df_train.shiptype
X_test=df_test.drop('shiptype', axis=1)
y_test=df_test.shiptype


#Creacion del modelo
pipe = Pipeline([('scaler', MinMaxScaler()),('classifier',RandomForestClassifier
(n_estimators=100))])
model = pipe.fit(X_train, y_train)
y_pred = model.predict(X_test)

#Resultados
print(classification_report(y_test, y_pred, zero_division=0))

forest = model['classifier']
importances = forest.feature_importances_
std = np.std([tree.feature_importances_ for tree in forest.estimators_], axis=0)
fi = pd.Series(importances, index=X_train.columns).reset_index()
fi.columns = ['att', 'importance']
fi = fi.sort_values(by='importance', ascending=False).reset_index(drop=True)
g = sns.barplot(data=fi, x='att', y='importance', color='b')
g.figure.set_size_inches(20, 5)
g.set_xticklabels(g.get_xticklabels(), rotation=90)

cur.close()
conn.close()
```

ANEXO VI: CÓDIGO *DATOS_MIXTO.IPYNB*

```
#Librerias
import pandas as pd
import sqlite3
import numpy as np
import seaborn as sns
from tqdm import tqdm
import matplotlib.pyplot as plt
import folium
import sqlalchemy
import h3

#Funciones predeterminadas
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import classification_report
from sklearn.ensemble import RandomForestClassifier
from sklearn.pipeline import Pipeline
from imblearn.over_sampling import SMOTE
from imblearn.over_sampling import RandomOverSampler
from imblearn.under_sampling import RandomUnderSampler

#Base de datos estaticos
path_data ='../CONJUNTO/estaticos_14dias.sql'
por = sqlite3.connect(path_data)
cur = por.cursor()

#Base de datos dinamicos
path_data = r'../CONJUTNO/dinamicos_14dias.sql'
conn = sqlite3.connect(path_data)
cur = conn.cursor()

#Juntar archivos
df1 = pd.read_sql('SELECT * FROM estaticos', con=por)
df1 = df1[df1.replace([np.inf, -np.inf], np.nan).notnull().all(axis=1)]
df1 = df1.dropna()
df1 = df1.drop_duplicates()

df2 = pd.read_sql('SELECT * FROM dinamicos', con=conn)
df2 = df2[df2.replace([np.inf, -np.inf], np.nan).notnull().all(axis=1)]
df2 = df2.dropna()
df2 = df2.drop_duplicates()

df = pd.merge(df1,df2, on=('imo','shiptype'))
df.to_sql('mixto', con=conn, if_exists='replace', index=False)

df = pd.read_sql("""SELECT imo, aol, ldivw, b, a, area, grith, len, ldivd, wid,
vs, wdivd,  aml, amt, h3_distanceFL, q50_sog, h3r7_stdfreq, draught, avg_sog,
h3r7_distanceFL, d, shiptype FROM static_and_dinamic""", con=conn)

#Dividir en train y test en base a imo
dfa = df.filter(items=['imo'])
dfa = dfa. drop_duplicates()

df_tr, df_ts = train_test_split(dfa, random_state = 42)
```

```
df_train = pd.merge(df_tr, df)
df_train = df_train.drop('imo', axis=1)
df_test = pd.merge(df_ts, df)
df_test = df_test.drop('imo', axis=1)

#Dividir para definir modelo
X_train=df_train.drop('shiptype', axis=1)
y_train=df_train.shiptype
X_test=df_test.drop('shiptype', axis=1)
y_test=df_test.shiptype
pipe = Pipeline([('scaler', MinMaxScaler()),('classifier',RandomForestClassifier
(n_estimators=100))])
model = pipe.fit(X_train, y_train)
y_pred = model.predict(X_test)

#Resultados
print(classification_report(y_test, y_pred, zero_division=0))

forest = model['classifier']
importances = forest.feature_importances_
std = np.std([tree.feature_importances_ for tree in forest.estimators_], axis=0)
fi = pd.Series(importances, index=X_train.columns).reset_index()
fi.columns = ['att', 'importance']
fi = fi.sort_values(by='importance', ascending=False).reset_index(drop=True)
g = sns.barplot(data=fi, x='att', y='importance', color='r')
g.figure.set_size_inches(8, 5)
g.set_xticklabels(g.get_xticklabels(), rotation=90)


cur.close()
conn.close()
por.close()
```