Utilización de Machine Learning para proporcionar a los deportistas acuáticos herramientas innovadoras para la selección del aparejo a utilizar.

Windcaddy Algorithm

Benjamín C. Sánchez La O

Business Intelligence

Nombre Tutor/a de TF
Humberto Andrés Sanz
Profesor/a responsable de la asignatura
Atanasi Daradoumis

16/06/2024

Universitat Oberta
de Catalunya

FICHA DEL TRABAJO FINAL

Título del trabajo:	Descripción del trabajo
Nombre del autor:	Benjamín Carmelo Sánchez La O
Nombre del consultor/a:	Humberto Andrés Sanz
Nombre del PRA:	Atanasi Daradoumis
Fecha de entrega (mm/aaaa):	06/2024
Titulación o programa:	Grado de Ingeniería informática
Área del Trabajo Final:	Business Intelligence
Idioma del trabajo:	Castellano
Palabras clave	#Windsurf #Caddie #Condiciones

Resumen del Trabajo

El presente trabajo se enfoca en el desarrollo de un algoritmo con Machine Learning (ML) llamado WindCaddy, destinado a mejorar la toma de decisiones en deportes acuáticos impulsados por el viento, como el windsurf. Utilizando datos meteorológicos proporcionados por la Agencia Estatal de Meteorología (AEMET), se implementan algoritmos de análisis predictivo para recomendar el equipo óptimo según las condiciones del viento y el agua. La metodología abarca desde la obtención y procesamiento de datos hasta la validación de los modelos desarrollados. Se utilizan técnicas de aprendizaje automático y se desarrollan algoritmos personalizados para la selección del aparejo adecuado para cada sesión. Los resultados muestran una mejora significativa en la precisión de las recomendaciones de equipo en comparación con métodos tradicionales. Se destacan los beneficios en términos de rendimiento y seguridad para los deportistas. En conclusión, WindCaddy ofrece una solución innovadora que fusiona la pasión por los deportes acuáticos con las tecnologías de análisis de datos, mejorando la experiencia de los deportistas y proporcionando herramientas para una toma de decisiones informada y optimizada.

Abstract

The present work focuses on the development of a Machine Learning (ML) based algorithm called WindCaddy, aimed at improving decision making in wind-driven water sports, such as windsurfing. Using meteorological data provided by the Spanish Meteorological Agency (AEMET), predictive analysis algorithms are implemented to recommend the optimal equipment according to wind and water conditions. The methodology ranges from data collection and processing to the validation of the developed models. Machine learning techniques are used and customised algorithms are developed for the

selection of the appropriate rig for each session. The results show a significant improvement in the accuracy of equipment recommendations compared to traditional methods. The benefits in terms of performance and safety for athletes are highlighted. In conclusion, WindCaddy offers an innovative solution that fuses a passion for water sports with data analytics technologies, enhancing the athlete experience and providing tools for informed and optimised decision making.

Índice

1. Introducción 1
1.1. Contexto y justificación del Trabajo 1
1.2. Objetivos del Trabajo 2
1.3. Impacto en sostenibilidad, ético-social y de diversidad 3
1.4. Enfoque y método seguido 3
1.5. Planificación del Trabajo 5
1.6. Breve sumario de productos obtenidos 12
1.7. Breve descripción de los capítulos de la memoria 13
2. Materiales y métodos 16
2.1. Selección de Herramientas y Software 16
2.2. Enfoque Metodológico 16
2.3. Valoración económica 17
3. Descripción del origen de los datos 19
3.1 Origen de los datos meteorológicos 19
3.2 Spots 20
4. Conclusiones Fase I 21
5. Proceso ETL 22
5.1 Parametrización del software 22
5.2 Modelo de datos y carga inicial 23
5.3 Carga desde OpenData AEMET 23
5.4 Carga desde Puertos.es 24
5.5 Conjunto de datos a etiquetar 24
6. Etiquetado de los datos 26
6.1 Tecnologías usadas y arquitectura 26
6.2 Promoción 27
7. Análisis exploratorio 28
7.1 Exploración del conjunto de datos 28
7.2 Preprocesamiento y gestión de características 32
8. Análisis de componentes principales 49
9. Conclusiones Fase II 56
10. Conjuntos *holdout set,* entrenamiento y prueba 58
10.1 Opciones del conjunto de datos 59
10.2 Validación cruzada 60
11. Selección del modelo 63
11.1 Modelos evaluados 63
11.2 Evaluación de los modelos 70
11.3 Selección de los modelos a usar 74
12. Ajustes del modelo 75
12.1 Ajuste Bosque aleatorio 75
12.2 Ajuste XGBRegressor 77
12.3 Conjunto final de hiperparámetros. 79
13. Desarrollo y test del modelo final 80
13.1 Algoritmo final 80
13.2 Extracción de la muestra de datos 80
13.3 Validación visual 81
14. Conclusiones Fase III 82
15. Validación 83

15.1 Consideraciones en la validación 83
15.2 Validación versión 1 86
15.3 Desarrollo de versiones v2 y v3 87
15.4 Validación versión 2 91
15.5 Validación versión 3 93
16. Resultados 95
16.1 Descripción de los datos 95
16.2 Vector de entrada (inputs) 95
16.3 Modelos Evaluados 95
16.4 Métricas de Evaluación 96
16.5 Resultados 96
16.6 Análisis de Resultados 96
17. Conclusiones Fase IV y trabajos futuros 98
17.1 Descripción de las conclusiones del trabajo 98
17.2 Consecución de los objetivos planteados 98
17.3 Metodología y seguimiento de la planificación 98
17.4 Evaluación de impactos 98
17.5 Líneas de trabajo futuro 99
18. Leyendas y tablas explicativas 101
12.1 Evaluación del riesgo 101
12.2 Leyenda estados de tareas 101
12.3 Leyenda estados de hitos 102
19. Glosario 103
20. Referencias 104
21. Anexos 107
21.1 Consulta de carga de datos 108
21.2 Código función windcaddyalg 109
21.3 Consulta de extracción de datos para validación visual 110
21.4 Relación de los paquetes Python usados 111
21.5 Ficheros Docker Formulario Etiquetado 112

Lista de figuras

Figura 1: Ejecución del proyecto (Gantt) 9
Figura 2: Hitos del proyecto (Gantt) 10
Figura 3: Correo recibido desde puertos.es 20
Figura 4: Flujo de trabajo Fase II [ETL] 22
Figura 5: Flujo de trabajo Fase II [ETIQUETADO] 26
Figura 6: Capturas del formulario de etiquetado. 26
Figura 7: Cartelería de promoción formulario web 27
Figura 8: Flujo de trabajo Fase II [ANALISIS EXPLORATORIO] 28
Figura 9: Frecuencias de las dimensiones numéricas 38
Figura 10: Frecuencias de las dimensiones no numéricas 40
Figura 11: Matriz de correlación Pearson de las variables continuas 42
Figura 12: Relación de valoración con algunas variables 44
Figura 13: Relación valoración por perfil con algunas variables 45
Figura 14: Flujo de trabajo Fase II [PCA] 49
Figura 15: Mapa calor carga variables (PCA0) 51
Figura 16: Varianza explicada acumulada (PCA0) 51
Figura 17: Relación variables con PC1 y PC2 (PCA0) 51
Figura 18: Datos PCA0 51
Figura 19: Mapa calor carga variables (PCA1) 52
Figura 20: Varianza explicada acumulada (PCA1) 52
Figura 21: Relación variables con PC1 y PC2 (PCA1) 52
Figura 22: Datos PCA1 52
Figura 23: Mapa calor carga variables (PCA2) 53
Figura 24: Varianza explicada acumulada (PCA2) 53
Figura 25: Relación variables con PC1 y PC2 (PCA2) 53
Figura 26: Datos PCA2 53
Figura 27: Mapa calor carga variables (PCA3) 54
Figura 28: Varianza explicada acumulada (PCA3) 54
Figura 29: Relación variables con PC1 y PC2 (PCA3) 54
Figura 30: Datos PCA3 54
Figura 31: Flujo de trabajo Fase III [PARTICIONADO] 58
Figura 32: Flujo de trabajo Fase III [ESTUDIO MODELOS] 63
Figura 33: Predicción [`valoracion`] métricas modelos 70
Figura 34: Predicción [`sail_size`] métricas modelos (Opc. 1) 71
Figura 35:Predicción [`wind_board_size`] métricas modelos (Opc. 1) 71
Figura 36: Predicción [sail_size] métricas modelos (Opc. 2) 72
Figura 37: Predicción [`wind_board_size`] métricas modelos (Opc. 2) 72
Figura 38: Comparativa modelos, opciones y métricas [`sail_size`] 73
Figura 39: Comparativa modelos, opciones y métricas [`wind_board_size`]. 73
Figura 40: Flujo de trabajo Fase III [AJUSTES DEL MODELO] 75
Figura 41: Resultados del ajuste para la variable `valoracion` y modelo `Random Forest`. 76
Figura 42: Resultados del ajuste para la variable `wind_board_size` y modelo `Random Forest`. 77
Figura 43: Resultados del ajuste para la variable `sail_size` y modelo `XGBRegressor`. 78

Figura 44:Flujo de trabajo Fase III [TEST MODELO] 80
Figura 45: Muestra de la validación visual de windcaddyalg 81
Figura 46: Resultados validacion windcaddyalg version 1 86
Figura 47: Métricas de valoración en los modelos categóricos evaluados 89
Figura 48: Matriz de confusión para valoración de los modelos evaluados 90
Figura 49: Resultados validación windcaddy version 2 91
Figura 50: Resultados validacion windcaddy version 3 93

1. Introducción

En el apasionante mundo de los deportes acuáticos impulsados por el viento, la capacidad de tomar decisiones informadas sobre el equipo a utilizar en función de las condiciones meteorológicas es esencial para optimizar el rendimiento y garantizar la seguridad de los deportistas. Sin embargo, la predicción precisa del aparejo adecuado para cada sesión puede ser un desafío, dado el dinamismo y la complejidad de los factores que influyen en las condiciones del viento y del agua.

En respuesta a esta necesidad, surge *WindCaddy*, un proyecto de Trabajo de Fin de Grado (TFG) que busca aprovechar las capacidades del *Machine Learning* (ML) para proporcionar a los deportistas acuáticos herramientas innovadoras para la selección del aparejo a utilizar. Este proyecto se enfoca en desarrollar una plataforma digital que integre datos meteorológicos con técnicas avanzadas de análisis de datos, con el objetivo de ofrecer pronósticos meteorológicos y sugerencias personalizadas sobre el equipo óptimo para cada sesión y deporte.

WindCaddy representa una innovadora propuesta que fusiona la pasión por los deportes acuáticos con las tecnologías en análisis de datos, con el propósito de mejorar la experiencia, seguridad y el rendimiento de los deportistas en el agua.

1.1. Contexto y justificación del Trabajo

Los deportes objetivo de este proyecto son *windsurf*, *kitesurf* y *wingfoil*. Todos ellos requieren de una combinación única de habilidades atléticas, conocimiento técnico y comprensión de las condiciones del entorno que, hasta ahora, sólo brinda la experiencia.

El equipo utilizado, incluidas las velas, tablas y *foils*, juega un papel fundamental en el rendimiento y la seguridad del deportista. La selección adecuada del equipo depende en gran medida de las condiciones meteorológicas, como la velocidad y dirección del viento, las corrientes, olas y hasta la temperatura del agua.

Tradicionalmente, estos deportistas han dependido de fuentes de información meteorológica general, como aplicaciones de pronóstico del tiempo, para tomar decisiones sobre qué equipo utilizar en sus sesiones. Sin embargo, estas fuentes pueden no proporcionar la precisión necesaria para las necesidades específicas de estos usuarios, lo que puede resultar en decisiones inadecuadas en la selección del aparejo o incluso peligrosas.

La utilización de *ML* con datos meteorológicos para sugerir el equipamiento adecuado en estos deportes ofrece una solución innovadora y altamente efectiva para este desafío. Al aplicar técnicas avanzadas de análisis de datos a conjuntos de datos meteorológicos históricos y en tiempo real, los deportistas pueden obtener pronósticos más precisos y detallados sobre las condiciones en el agua lo que permitirá al sistema sugerir que material es el óptimo de acuerdo a las preferencias y disponibilidad del material del usuario.

La plataforma *WindCaddy* tiene como objetivo ofrecer a los deportistas acuáticos una herramienta completa que combina datos meteorológicos con la evaluación del usuario sobre las recomendaciones de equipo. Esta integración permite que el algoritmo se retroalimente continuamente, mejorando así la precisión y relevancia de las sugerencias proporcionadas. Esta aplicación no solo tiene el potencial de mejorar el rendimiento deportivo al proporcionar recomendaciones precisas de equipo, sino que también puede aumentar la seguridad al ayudar a los deportistas a evitar condiciones peligrosas en el agua.

Además, *WindCaddy* tiene la capacidad de personalizar sus sugerencias según los pronósticos, preferencias, deporte, modalidad (*waves*, *slalom*, *freestyle*…) y habilidades individuales de cada deportista, lo que lo convierte en una herramienta altamente adaptable y relevante para la comunidad de deportes acuáticos impulsados por el viento.
En resumen, la utilización de *BI* representa una innovación significativa que puede mejorar tanto el rendimiento deportivo como la seguridad de estos deportistas.

1.2. Objetivos del Trabajo

El objetivo principal de este TFG es el desarrollo de un motor de sugerencias basadas en las predicciones meteorológicas y datos históricos que oferte de forma personalizada la opción óptima (o la mejor entre las disponibles) de material a utilizar basándose en los diferentes *inputs* parametrizados.
Los objetivos específicos son los siguientes:

1. Recopilación de Datos
Recolectar datos meteorológicos históricos y en tiempo real de fuentes confiables y relevantes para los deportes acuáticos impulsados por el viento, como la velocidad y dirección del viento, corrientes, olas, temperatura del agua, entre otros.

2. Análisis Predictivo
Aplicar técnicas avanzadas de análisis predictivo a los datos recopilados para identificar patrones y tendencias en las condiciones meteorológicas y su impacto en el rendimiento deportivo.

3. Desarrollo de Algoritmos
Diseñar algoritmos personalizados que utilicen los datos meteorológicos y el análisis predictivo para recomendar el equipo óptimo (vela, tabla, foil, etc.) para cada sesión de forma personalizada al usuario.

4. Validación
Validar de forma adecuada las sugerencias aportadas por el algoritmo.

Al alcanzar estos objetivos, *WindCaddy* se convertiría en una herramienta de alto valor para sus usuarios, proporcionándoles información precisa y personalizada que les permita optimizar su rendimiento y seguridad en el agua.

1.3. Impacto en sostenibilidad, ético-social y de diversidad

El desarrollo y la implementación de *WindCaddy* tienen el potencial de generar impactos significativos en las dimensiones de sostenibilidad, ético-social y diversidad.

1.3.1 Dimensión de sostenibilidad

Reducción de la huella ecológica:
Al proporcionar recomendaciones precisas sobre el equipo a utilizar según las condiciones meteorológicas, se puede minimizar el uso innecesario de material y recursos, lo que contribuye a la sostenibilidad medioambiental.

Promoción de la economía circular
Al sugerir la reutilización de equipo existente en lugar de la adquisición de nuevo material, *WindCaddy* puede fomentar prácticas más sostenibles dentro de la comunidad de deportes acuáticos impulsados por el viento.

1.3.2 Dimensión ético-social

Mejora del rendimiento y seguridad
Al proporcionar recomendaciones precisas y personalizadas, *WindCaddy* puede ayudar a los deportistas a optimizar su rendimiento y evitar situaciones de riesgo en el agua, lo que promueve la seguridad y el bienestar de la comunidad deportiva.

Acceso equitativo a la información
Al ofrecer pronósticos meteorológicos y sugerencias de equipo de manera accesible y personalizada, *WindCaddy* puede reducir las barreras de acceso a información crucial para la práctica de los deportes acuáticos, promoviendo así la inclusión y la equidad en la comunidad deportiva.

1.3.3 Dimensión de diversidad

Adaptabilidad a diversas necesidades y preferencias
Al permitir la personalización de las recomendaciones según las preferencias individuales, habilidades y necesidades de cada deportista, *WindCaddy* puede atender a la diversidad dentro de la comunidad de deportes acuáticos, promoviendo así la inclusión y la diversidad de género, habilidades y preferencias en la práctica deportiva.

1.4. Enfoque y método seguido

Para mantener un enfoque práctico y realista dentro del alcance de este TFG, se limitará la obtención de datos para el entrenamiento del algoritmo a un área específica: la isla de Gran Canaria. Además, en esta primera fase de desarrollo,

se restringirá el enfoque del algoritmo exclusivamente al windsurf como deporte principal, sugiriendo volumen de tabla y tamaño de vela.

Inicialmente, el algoritmo no considerará los diversos modelos de tablas, velas o quillas. Al excluir inicialmente los diferentes modelos de tablas, velas y quillas del alcance del algoritmo, se puede mantener un enfoque más dirigido y manejable en la predicción del aparejo, lo que facilita el desarrollo, la implementación y la validación de este. Esto permitirá que el algoritmo proporcione recomendaciones útiles y precisas a los usuarios, sin comprometer la calidad o la eficacia del sistema en su conjunto.

Los datos meteorológicos se recopilarán utilizando la API de AEMET OpenData y los datos solicitados a Puertos del Estado, específicamente de las estaciones meteorológicas y puntos del modelo ubicados próximos a Gran Canaria. Los datos recogidos se asociarán a uno o varios *spots* de la isla según su proximidad geográfica y conveniencia. La lista de *spots* en la isla se obtendrá del siguiente enlace: Spots Gran Canaria.

Para obtener una primera etiqueta de una parte de los datos, se consultará a varios deportistas, tanto profesionales como aficionados. Se les proporcionarán los datos de entrada del algoritmo y se les pedirá que sugieran el equipo que consideren adecuado para ese spot y condiciones específicas, según la modalidad a practicar.

Se empleará una técnica de aprendizaje **semi-supervisado**, dado que es adecuada cuando solo se dispone de etiquetas para una pequeña parte de los datos. Este modelo aprende tanto de los datos con etiquetas como de aquellos sin etiquetar para realizar predicciones sobre nuevos datos.

Con respecto a la metodología a utilizar, se debate entre la metodología en cascada y una metodología ágil. El siguiente cuadro comparativo muestra sus principales diferencias:

Aspecto	*Metodología en Cascada*	*Metodología Ágil*
Estructura	Lineal y secuencial	Adaptable y flexible
Enfoque	Detallado en cada fase	Iterativo e incremental
Planificación	Requiere planificación detallada desde el principio	Se adapta a medida que avanza el proyecto
Flexibilidad	Menos flexible, cambios difíciles una vez iniciado	Altamente flexible, permite ajustes y cambios
Entrega de Resultados	Entrega al final del proyecto	Entrega continua de resultados tangibles
Colaboración	Menos énfasis en la colaboración	Fomenta la colaboración cercana entre los miembros del equipo y los interesados
Adaptabilidad	Menor adaptabilidad a cambios en requisitos	Mayor adaptabilidad a cambios en requisitos
Gestión de Riesgos	Identificación temprana y mitigación de riesgos	Enfoque en la gestión continua de riesgos

Aunque la metodología ágil sería ideal para proyectos de este tipo, especialmente fuera del ámbito académico y con un cliente involucrado, las restricciones en el alcance mencionadas anteriormente establecen un marco claro en cuanto a los requisitos desde el inicio del proyecto.

Por lo tanto, una metodología en cascada que permita definir de manera secuencial y clara los objetivos de cada fase, con cierta holgura en la programación de hitos que puedan absorber las desviaciones de cada fase.

1.5. Planificación del Trabajo

1.5.1 Roles identificados

Para acometer el proyecto con garantías idealmente serían necesarios los siguientes roles:

Rol	Descripción	Salario	Coste
Científico de datos (CD)	Responsable del desarrollo de algoritmos predictivos y del análisis de datos para proporcionar recomendaciones de equipo óptimo	19,23 €/Hora	25,96 €/Hora
Ingeniero de datos (ID)	Encargado de la extracción, transformación y carga (ETL) de los datos, así como del diseño y mantenimiento de la infraestructura de almacenamiento de datos.	19,23 €/Hora	25,96 €/Hora
Especialista en meteorología (MET)	Colaborador clave en la validación de los datos meteorológicos y en la comprensión de su impacto en los deportes acuáticos impulsados por el viento	21,63 €/Hora	29,20 €/Hora
Gerente de proyecto (PM)	Responsable de la planificación, coordinación y supervisión general del proyecto, asegurando que se cumplan los plazos y objetivos.	24,04 €/Hora	32,45 €/Hora

Las fuentes de consulta para los salarios por perfil se han obtenido desde los recursos web que se relacionan en la bibliografía de este documento [1] [2] [3].

Por otro lado, en el cálculo del coste, se ha tenido en cuenta que, en España, los costos adicionales asociados con la contratación de empleados pueden oscilar entre un 30% y un 40% del salario bruto, lo que incluye los pagos de seguridad social, desempleo, y otros impuestos. Sin embargo, esto puede variar dependiendo de diversos factores, como el tipo de contrato, la industria y la ubicación.

Se considera un promedio del 35% en costos adicionales sobre el salario bruto, por lo que para calcular el costo real de la persona se obtiene la siguiente fórmula:

$$CostoRecurso = SalarioBruto + (SalarioBruto * 1{,}35)$$

1.5.2 Análisis de riesgos

Código	Nombre	Causa	Descripción	Consecuencia	Probabilidad	Impacto	Nivel
R01	Datos insuficientes	No es posible obtener datos suficientes	No se han podido obtener datos de fuentes confiables o estos son insuficientes	Imposibilidad/Retrasos en la consecución del proyecto	Bajo	Alto	Mediano
R02	Etiquetado insuficiente	No se ha podido obtener el etiquetado para los datos del proyecto	Los usuarios consultados para etiquetar los datos no han podido/querido hacerlo.	Pobreza en el modelo predictivo y sugerencias sesgadas	Bajo	Alto	Mediano
R03	Retrasos en las fases	Acumulación de deuda técnica	No se ha podido acometer una o varias tareas técnicas por falta de tiempo o capacidad en el plazo establecido.	Retrasos en las tareas subsecuentes e hitos.	Bajo	Mediano	Mediano
R04	ETL complejo	Complejidad elevada para la limpieza y preparación de datos	El proceso de ETL implica desafíos técnicos y operativos significativos debido a la complejidad y variedad de las fuentes de datos utilizadas.	Una complejidad demasiado elevada podría retrasar el desarrollo, afectar a la precisión de los algoritmos y aumentar el riesgo de errores.	Bajo	Mediano	Mediano
R05	Desarrollo complejo	Dificultades en el desarrollo de los diferentes algoritmos	Posibles dificultades para el desarrollo debido a la complejidad de este.	Podría impactar en la precisión de las recomendaciones de equipo, retrasar la implementación de WindCaddy y afectar la	Bajo	Mediano	Mediano

				satisfacción de los usuarios finales.			
R06	HW insuficiente	La infraestructura de la que se dispone no es suficiente.	Es posible que el volumen de datos y/o procesamiento necesario exceda de las capacidades del hardware que se dispone.	Imposibilidad de realizar las tareas de forma óptima en los plazos planificados.	Bajo	Alto	Mediano
R07	Validación incorrecta	La validación del algoritmo no es adecuada o insuficiente	La validación no se ha realizado correctamente o el algoritmo no satisface unos mínimos.	El algoritmo no dará resultados útiles al usuario.	Bajo	Alto	Mediano

Véase Evaluación del riesgo

1.5.3 Plan de contingencia

Siguiendo un enfoque y estrategia proactivo, se confecciona el siguiente plan para la prevención de riesgos donde figuran todas las actividades y acciones necesarias y posibles en el plan de proyecto, para minimizar el impacto o prevenir los riesgos. Se definen las acciones a tomar como contingencia para la mitigación del impacto de un riesgo materializado. Se aprovisionan recursos económicos, para afrontar estas contingencias.

Medidas correctoras					
Código	**Riesgo**	**Acción**	**Tipo**	**Riesgo Residual**	**Fecha límite**
A01	R01	Obtener datos de fuentes y webs alternativas, como por ejemplo NOAA.	Correctora	Bajo	18/03/2024
A02	R02	Lanzar una segunda ronda de consultas a los usuarios del piloto para aumentar el número de etiquetas en el *dataset*	Correctora	Bajo	03/04/2024
A03	R03	Aumentar el número de horas semanales dedicadas al proyecto.	Correctora	Bajo	N/A
A04	R04	Implementar procesos automatizados de limpieza y normalización de datos para garantizar la consistencia y calidad de los datos.	Correctora	Bajo	27/03/2024
A05	R05	Soporte de parte del profesor consultor y el foro del aula.	Correctora	Bajo	N/A
A06	R06	Búsqueda de alternativas en *cloud*	Correctora	Bajo	N/A
A07	R07	Revisión del proceso de validación que incluya más pruebas comparativas y más retroalimentación de los usuarios del piloto	Correctora	Bajo	02/06/2024

1.5.4 Planificación temporal (Tareas)

Nombre	Fecha de inicio	Fecha de fin
Ejecución Proyecto	28/2/24	16/6/24
Propuesta TFG	28/2/24	3/3/24
Busqueda de tema para TFG	28/2/24	29/2/24
Preparación Propuesta	1/3/24	1/3/24
Envío Propuesta	2/3/24	2/3/24
Aprobación Propuesta	3/3/24	3/3/24
Fase I	4/3/24	14/3/24
Análisis preliminar	4/3/24	5/3/24
Valoracion económica	6/3/24	8/3/24
Identifcación de riesgos	9/3/24	10/3/24
Planificación Temporal	11/3/24	14/3/24
Fase II	15/3/24	17/4/24
Obtención de datos	15/3/24	18/3/24
Desarrollo ETL	19/3/24	27/3/24
Etiquetado Inicial	28/3/24	3/4/24
PCA	4/4/24	16/4/24
Dataset Final	17/4/24	17/4/24
Fase III	18/4/24	19/5/24
División Dataset	18/4/24	22/4/24
Estudio de modelos	23/4/24	27/4/24
Selección del modelo a usar	28/4/24	1/5/24
Desarrollo y test del modelo	2/5/24	19/5/24
Fase IV	20/5/24	16/6/24
Validación del modelo	20/5/24	29/5/24
Conclusiones finales	30/5/24	2/6/24
Documentación final	3/6/24	12/6/24
Entrega y cierre proyecto	13/6/24	16/6/24

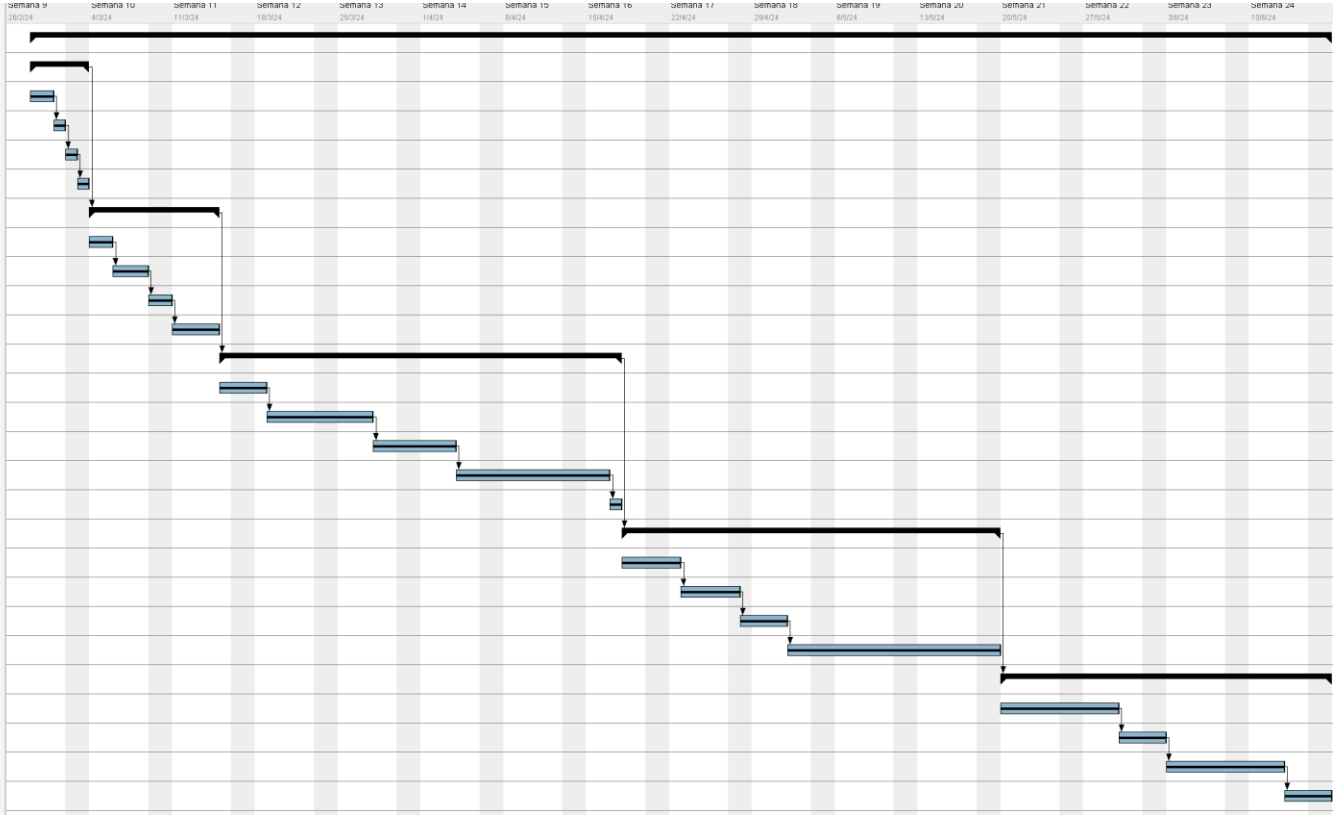


Figura 1: Ejecución del proyecto (Gantt)

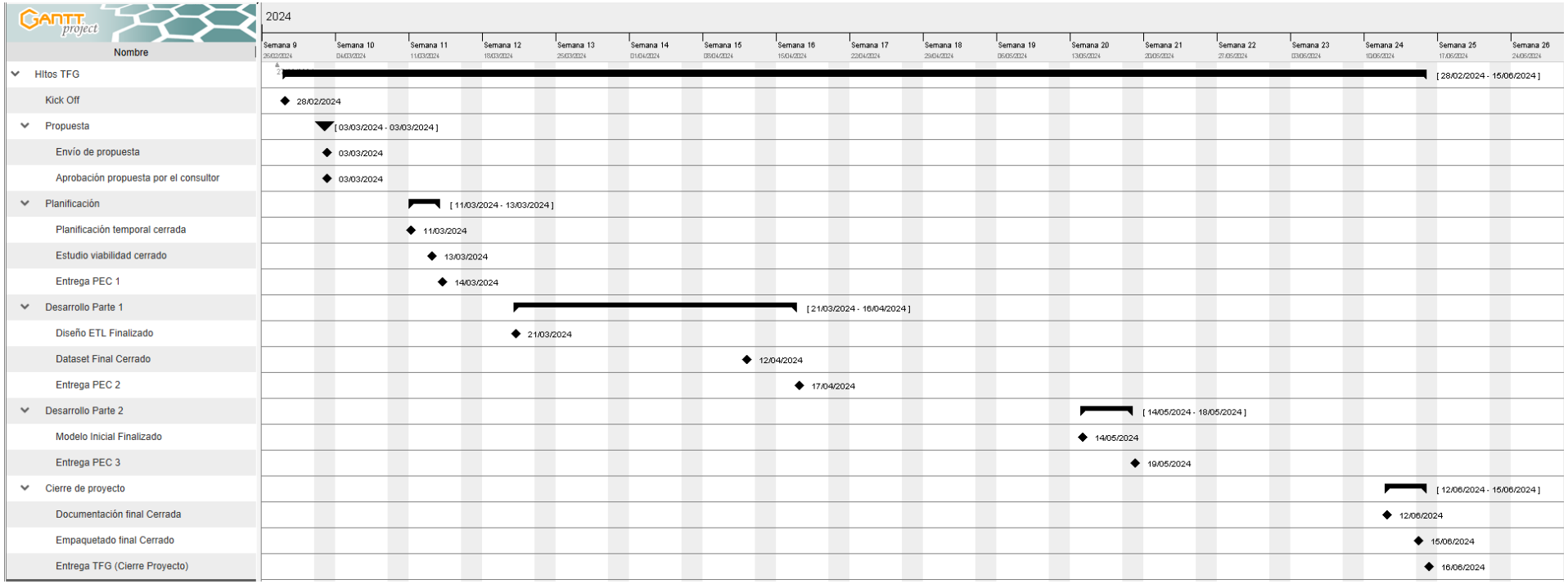


Figura 2: Hitos del proyecto (Gantt)

1.5.5 Relación de tareas

Tareas del proyecto					
Id	*Tarea*	**Roles**	**Fecha inicio**	**Fecha Fin**	**Estado**
1	**Propuesta TFG**		**28/02/2024**	**03/03/2024**	**Finalizada**
1.1	Búsqueda de tema TFG	-	28/02/2024	29/02/2024	Finalizada
1.2	Preparación Propuesta	-	01/03/2024	01/03/2024	Finalizada
1.3	Envío Propuesta	-	02/03/2024	02/03/2024	Finalizada
1.4	Aprobación Propuesta	-	03/03/2024	03/03/2024	Finalizada
2	**Fase I**	**CD/PM/MET**	**04/03/2024**	**14/03/2024**	**Finalizada**
2.1	Análisis preliminar	CD/MET	04/03/2024	05/03/2024	Finalizada
2.2	Valoración económica	PM	06/03/2024	08/03/2024	Finalizada
2.3	Identificación de riesgos	PM	09/03/2024	10/03/2024	Finalizada
2.4	Planificación Temporal	PM	11/03/2024	14/03/2024	Finalizada
3	**Fase II**	**ID/CD**	**15/03/2024**	**15/04/2024**	**Finalizada**
3.1	Obtención de datos	ID	10/03/2024	13/03/2024	Finalizada
3.2	Desarrollo ETL	ID	14/03/2024	22/03/2024	Finalizada
3.3	Etiquetado Inicial	CD	23/03/2024	29/03/2024	Finalizada
3.4	PCA	CD	30/03/2024	11/04/2024	Finalizada
3.5	Dataset Final	ID	12/04/2024	15/04/2024	Finalizada
4	**Fase III**	**CD**	**16/04/2024**	**17/05/2024**	**Finalizada**
4.1	División Dataset	CD	16/04/2024	20/04/2024	Finalizada
4.2	Estudio de modelos	CD	21/04/2024	25/04/2024	Finalizada
4.3	Selección modelo a usar	CD	26/04/2024	29/04/2024	Finalizada
4.4	Desarrollo y test modelo	CD	30/04/2024	17/05/2024	Finalizada
5	**Fase IV**	**ID/CD/PM**	**18/05/2024**	**15/06/2024**	**Finalizada**
5.1	Validación del modelo	CD/MET	18/05/2024	27/05/2024	Finalizada
5.2	Conclusiones Finales	ID/CD/PM	28/05/2024	31/05/2024	Finalizada
5.3	Documentación final	ID/CD	01/06/2024	10/06/2024	Finalizada
5.4	Entrega y cierre proyecto	PM	11/06/2024	15/06/2024	Finalizada

Véase Leyenda de estados de tareas

1.5.6 Hitos

Hitos internos del proyecto y observaciones		
Fecha	**Descripción**	**Estado**
28/02/2024	Kick Off	Alcanzado
03/03/2024	Envío de propuesta	Alcanzado
03/03/2024	Aprobación de propuesta por el consultor	Alcanzado
11/03/2024	Planificación temporal cerrada	Alcanzado
13/03/2024	Estudio de viabilidad cerrado	Alcanzado
14/03/2024	Entrega PEC 1	Alcanzado
21/03/2024	Diseño ETL Finalizado	Alcanzado
12/04/2024	Dataset Final Cerrado	Alcanzado
17/04/2024	Entrega PEC 2	Alcanzado
14/05/2024	Modelo inicial Finalizado	Alcanzado
19/05/2024	Entrega PEC 3	Alcanzado
12/06/2024	Documentación final Cerrada	Alcanzado

15/06/2024	Empaquetado final Cerrado	Alcanzado
16/06/2024	Entrega TFG (Cierre Proyecto)	Alcanzado

Véase leyenda de estados de hitos

Siguiendo la planificación propuesta el proyecto se cierra el 16/06/2024, en principio a tiempo para la entrega de la documentación y empaquetado de código. La holgura de la planificación es aceptable ya que finaliza a la par con el último *deadline.*

Por otro lado, el camino crítico lo compone la obtención de datos fiables y útiles para el modelo, así como la obtención del etiquetado inicial del mismo por parte de los participantes en el proyecto, asimismo, la documentación relativa al mismo se realiza en paralelo al desarrollo del algoritmo.

1.6. Breve sumario de productos obtenidos

Al final de este proyecto, los entregables serán los listados a continuación:

1. **Memoria del proyecto (este documento):**
   La memoria del proyecto es un documento que detalla todos los aspectos relevantes del proyecto, incluyendo los objetivos, la metodología utilizada, los resultados obtenidos, las conclusiones y cualquier otra información relevante. Sirve como registro completo y detallado del trabajo realizado durante el proyecto.

2. **Los algoritmos desarrollados:**
   Estos son los algoritmos específicos creados para analizar y procesar los datos recopilados, con el objetivo de alcanzar los objetivos del proyecto. Pueden incluir algoritmos de análisis predictivo, de clasificación, de regresión u otros tipos, según las necesidades del proyecto.

3. **Código ETL para la obtención de datos meteorológicos:**
   Este es el código desarrollado para extraer, transformar y cargar (ETL) los datos meteorológicos desde las fuentes consultadas. Este código es crucial para la recopilación y preparación de los datos necesarios para el análisis.

4. **Código fuente y ficheros SQL formulario web:**
   El etiquetado de datos se realizará manualmente mediante este formulario.

5. **El conjunto de datos usado:**
   Este conjunto de datos consiste en la información recopilada y preparada para su análisis. Puede incluir datos meteorológicos u otros tipos de datos relevantes para el proyecto.

6. **Resultados obtenidos del análisis y de la validación:**
   Estos resultados son el producto del análisis de los datos y la validación de los modelos y algoritmos desarrollados. Pueden incluir estadísticas, gráficos, conclusiones y cualquier otra información relevante que demuestre el éxito o los hallazgos del proyecto.

7. **Video presentación:**
   Este es un video que resume y presenta los aspectos más importantes del proyecto, incluyendo los objetivos, la metodología, los resultados y las conclusiones. Sirve como una forma efectiva de comunicar los hallazgos del proyecto de manera visual y accesible.

1.7. Breve descripción de los capítulos de la memoria

Capítulo 1: Introducción
Este capítulo establece el contexto y la justificación del proyecto, delineando objetivos específicos y considerando su impacto en sostenibilidad y diversidad.

Capítulo 2: Materiales y Métodos
En este capítulo se detallan los materiales utilizados en el proyecto, así como los métodos y técnicas empleadas para llevar a cabo las diferentes fases de trabajo. Se abordan aspectos como la selección de herramientas, software, y protocolos de trabajo, así como el enfoque metodológico y la planificación del trabajo, anticipando los productos obtenidos.

Capítulo 3: Descripción del Origen de los Datos
Este capítulo se centra en presentar el origen y naturaleza de los datos utilizados en el proyecto. Se incluyen detalles sobre los datos meteorológicos obtenidos y la información sobre los *spots* analizados.

Capítulo 4: Conclusiones Fase I
En este capítulo se resumen los principales hallazgos, conclusiones y lecciones aprendidas durante la Fase I del proyecto, proporcionando una evaluación integral del trabajo realizado hasta el momento

Capítulo 5: Proceso ETL
En este capítulo se describe el proceso de Extracción, Transformación y Carga (ETL) de los datos. Se explican los pasos seguidos para la parametrización del software, el modelo de datos, y la carga de datos desde diferentes fuentes como OpenData AEMET y Puertos.es.

Capítulo 6: Etiquetado de los Datos
Se aborda el proceso de etiquetado de los datos, incluyendo las tecnologías utilizadas, la arquitectura implementada, y el proceso de etiquetado en sí mismo.

Capítulo 7: Análisis Exploratorio
En este capítulo se lleva a cabo un análisis exploratorio de los datos, incluyendo la exploración del conjunto de datos, el preprocesamiento y la gestión de características.

Capítulo 8: Análisis de Componentes Principales (PCA)
Se presenta el análisis de Componentes Principales (PCA), una técnica utilizada para la reducción de la dimensionalidad de los datos y la identificación de patrones y estructuras subyacentes.

Capítulo 9: Conclusiones Fase II
En este capítulo se resumen los principales hallazgos, conclusiones y lecciones aprendidas durante la Fase II del proyecto, proporcionando una evaluación integral del trabajo realizado hasta el momento

Capítulo 10: Conjuntos de entrenamiento y prueba
Detalla la división de los datos en conjuntos de entrenamiento y prueba para el desarrollo y evaluación de modelos. Se evalúan estrategias de partición de datos y consideraciones sobre el equilibrio entre ambos conjuntos.

Capítulo 11: Selección del modelo
Se centra en la elección del modelo más adecuado para abordar el problema planteado en el proyecto. Incluye comparaciones entre diferentes algoritmos y técnicas de modelado, así como criterios de selección.

Capítulo 12: Ajustes del modelo
Describe los procesos de ajuste de hiperparámetros y optimización del modelo seleccionado. Incluye técnicas como búsqueda de hiperparámetros, validación cruzada y ajuste fino del modelo.

Capítulo 13: Desarrollo y test del modelo final
Detalla el proceso de desarrollo del modelo final, utilizando los conjuntos de entrenamiento y prueba definidos anteriormente. Incluye una primera validación visual del algoritmo.

Capítulo 14: Conclusiones Fase III
Resume los principales hallazgos, conclusiones y lecciones aprendidas durante la Fase III del proyecto, ofreciendo una evaluación integral del trabajo realizado hasta ese momento haciendo énfasis en las opciones seleccionadas en cada apartado anterior.

Capítulo 15: Validación
Este capítulo se centra en la validación del proyecto. Comienza con consideraciones generales sobre la validación y luego describe el proceso de validación en sus diferentes versiones (v1, v2 y v3).

Capítulo 16: Resultados:
Aquí se presentan los resultados obtenidos del análisis y evaluación de los modelos desarrollados en el proyecto. Se describe el conjunto de datos utilizado, los vectores de entrada, los modelos evaluados, las métricas de evaluación y los resultados obtenidos. También se realiza un análisis de los resultados.

Capítulo 17. Conclusiones y trabajos futuros:
En este capítulo se resumen las conclusiones del proyecto, se evalúa la consecución de los objetivos planteados, se analiza la metodología y el seguimiento de la planificación, se evalúan los impactos y se proponen líneas de trabajo futuro.

Capítulo 18. Leyendas y tablas explicativas:
Este capítulo proporciona una serie de leyendas y explicaciones para facilitar la comprensión de las tablas y figuras presentadas en el documento. Incluye la evaluación del riesgo, leyendas de estados de tareas y de hitos.

Capítulo 19. Glosario:
Aquí se encuentran definiciones de términos técnicos y específicos utilizados en el proyecto.

Capítulo 20. Referencias:
Se listan todas las fuentes y referencias bibliográficas utilizadas en el proyecto.

Capítulo 21. Anexos:
Este capítulo contiene información adicional relevante para el proyecto que complementa el contenido principal del documento.

2. Materiales y métodos

En este capítulo, se detallan los materiales utilizados en el proyecto, así como los métodos y técnicas empleadas para llevar a cabo las diferentes fases de trabajo. Se abordan aspectos como la selección de herramientas, software y protocolos de trabajo, así como el enfoque metodológico y la planificación del trabajo, anticipando los productos obtenidos.

2.1. Selección de Herramientas y Software

Para el desarrollo del proyecto *WindCaddy*, se han seleccionado diversas herramientas y software que se ajustan a las necesidades específicas de cada etapa.

Se ha empleado código `Python` con el fin de desarrollar el proceso `ETL` (Extracción, Transformación y Carga), utilizando las bibliotecas `` `requests` `` para extraer datos a través de `REST` y `` `psycopg2` `` para establecer conexión y persistir la información en una base de datos relacional `PostgreSQL`, la cual está alojada en un contenedor `Docker`. Además, se emplea `SQL` para llevar a cabo la manipulación y consulta de los datos.

Para llevar a cabo el etiquetado, se ha elegido un enfoque que implica un *frontend* alojado en un servidor `Nginx` (también en `Docker`), desarrollado con `React`, y un *backend* que proporciona operaciones `REST`, construido sobre `Spring Boot`/`Java`. Este *backend* se encarga de persistir la información en la base de datos `PostgreSQL` mencionada anteriormente. Como plataforma de virtualización se ha usado `VMWare Player` en su versión gratuita.

En cuanto al procesamiento de datos y análisis predictivo, se ha optado por utilizar `Python` como lenguaje de programación principal, debido a su versatilidad y las bibliotecas disponibles para el análisis de datos, como `Pandas`, `NumPy` y `Scikit-learn` entre otros. El listado completo de los paquetes usados puede consultarse en el anexo 21.4

2.2. Enfoque Metodológico

Aunque se definieron objetivos y entregables claros desde el inicio del proyecto, con la elección inicial de una metodología en cascada, se ha permitido cierta flexibilidad para adaptarse a los cambios y desafíos surgidos durante el desarrollo.

Esta combinación de la metodología en cascada con un grado de flexibilidad se ve respaldada por el *feedback* periódico recibido en las entregas programadas, donde el tutor evalúa el progreso. La decisión se fundamentó en la necesidad de adaptabilidad frente a los riesgos que se podrían materializar.

2.3. Valoración económica

Valoración económica (Personal)				
Id. Tarea	***Roles***	***Horas***	***Hora***	***Coste total (€)***
2.1	CD	20	25,96 €	519,20 €
	MET	5	29,20 €	146,00 €
2.2	PM	15	32,45 €	486,75 €
2.3	PM	10	32,45 €	324,50 €
2.4	PM	13	32,45 €	421,85 €
Totales FASE I	**Horas: 63**		**Coste: 1.898,30 €**	
3.1	ID	17	19,23 €	326,91 €
3.2	ID	25	19,23 €	480,75 €
3.3	CD	15	19,23 €	288,45 €
3.4	CD	23	19,23 €	442,29 €
3.5	ID	13	19,23 €	249,99 €
Totales FASE II	**Horas: 93**		**Coste: 1.788,39 €**	
4.1	CD	10	19,23 €	192,30 €
4.2	CD	20	19,23 €	384,60 €
4.3	CD	10	19,23 €	192,30 €
4.4	CD	30	19,23 €	576,90 €
Totales FASE III	**Horas: 70**		**Coste: 1.346,10 €**	
5.1	CD	20	19,23 €	384,60 €
	MET	5	29,20 €	146,00 €
5.2	ID	4	19,23 €	76,92 €
	CD	7	19,23 €	134,61 €
	PM	2	32,45 €	64,90 €
5.3	CD	20	19,23 €	384,60 €
5.4	PM	10	32,45 €	324,50 €
Totales FASE IV	**Horas: 68**		**Coste: 1.516,13 €**	

El costo económico en cuanto a recursos humanos asciende a un total de **6.548,92 €** con un coste temporal de **294 Horas, un coste hora promedio de 22,28 € aprox.**

Dada la índole académica del presente proyecto, todas las responsabilidades serán asumidas por el estudiante firmante. Sin embargo, para asegurar la viabilidad y eficacia del proyecto, se reconoce la necesidad de la colaboración en distintos roles. Idealmente, se requeriría la participación de profesionales en diversas áreas, además de posiblemente un ingeniero especializado en infraestructura en la nube para mantener y alojar la infraestructura tecnológica requerida.

Se excluye la evaluación económica de la infraestructura, dado que el proyecto se llevará a cabo en el equipo personal del estudiante que suscribe, el cual posee las siguientes especificaciones técnicas:

HW	Modelo
CPU	11th Gen Intel(R) Core(TM) i7-11850H @ 2.50GHz 2.50 GHz (8 núcleos y 16 hilos)
RAM	32GB (2 x M471A2G43BB2-CWE de 16GB cada una)
HDD	NVMe PC711 NVMe SK hynix 1TB

En cuanto a la valoración de licencias usadas durante el desarrollo del proyecto se obtiene lo siguiente:

Tipo	Fabricante	Software	Versión	Licencia	Coste (€)
OS	Microsoft	Windows 10 Pro	22H2	Incluido en el dispositivo	0,00
3GL	Python Software Foundation	Python	3.11	Licencia PSFL	0,00
IDE	JetBrains	PyCharm	2023.3.3	Professional Ed. (UOC)	0,00
IDE	JetBrains	IntelliJ	2023.3	Professional Ed. (UOC)	
SGDB	PGDG	PostgreSQL	APD[1]	PostgreSQL	0,00
3GL		React			
3GL		Java			
Soft. Containers	Docker Inc	Docker Desktop	APD[1]	Apache 2.0	0,00
Soft	VMWare	Player	17	Non-comercial	0,00

Considerando lo mencionado anteriormente, se estima que el coste aproximado para llevar a cabo el presente Trabajo de Fin de Grado (TFG) es de **338 horas**, con un margen de seguridad del **15%** para mitigar posibles desviaciones. Esto equivale a unos **7.530,64 €**, basándose en el coste promedio por hora calculado previamente.

[1] Aún por determinar

3. Descripción del origen de los datos

Se ha procedido a la selección de datos provenientes de diversas fuentes, las cuales son citadas más adelante. Estos datos serán unificados para formar un conjunto preliminar coherente, que será sujeto a un proceso de etiquetado manual por parte de los deportistas. Este etiquetado se llevará a cabo mediante un formulario en línea, y los datos resultantes constituirán el conjunto final a utilizar.

3.1 Origen de los datos meteorológicos

El origen de los datos provendrá principalmente de las siguientes fuentes que se listan en los siguientes subapartados:

3.1.1 AEMET OpenData

AEMET OpenData es una API REST desarrollado por AEMET que permite la difusión y la reutilización de la información meteorológica y climatológica de la Agencia Estatal de Meteorología.

AEMET OpenData permite la descarga gratuita de numerosos datos meteorológicos mediante consultes REST a los recursos publicados en:

- https://opendata.aemet.es/AEMET_OpenData_specification.json

Para acceder a estos recursos es necesario obtener una API KEY, desde la página principal de AEMET OpenData introduciendo el correo electrónico.

3.1.2 Puertos.es

Puertos estatales del Estado ofrece la posibilidad de descargar series temporales de parámetros de viento y oleaje procedentes de un modelado numérico sobre un punto (punto SIMAR).

Los puntos SIMAR, pertenecientes a Puertos del Estado (Ministerio de Fomento), comprenden series temporales de parámetros de viento y oleaje procedentes de modelado numérico con cadencia horaria y que se extiende desde 1958 hasta la actualidad. [4]

Para la obtención de estos datos es necesario solicitar la descarga de los parámetros y puntos SIMAR necesarios, mediante la aplicación puesta a tal efecto en:

- https://www.puertos.es/es-es/oceanografia/Paginas/portus.aspx

Una vez aprobada la descarga, se recibe un correo electrónico con los enlaces de descarga en formato CSV.

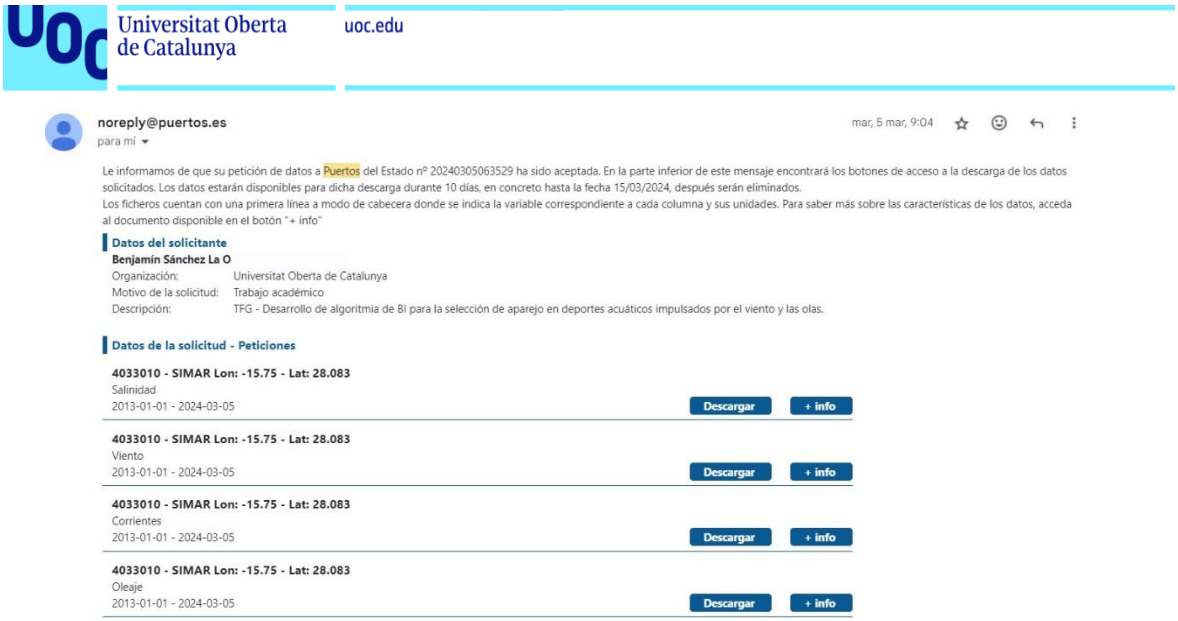


Figura 3: Correo recibido desde puertos.es

3.2 Spots

Los lugares seleccionados para este estudio se limitan a los *spots* más destacados de la isla de Gran Canaria para la práctica del windsurf. Esta selección se basa en la información recopilada de un centro deportivo local especializado en deportes acuáticos, la cual se puede consultar en los siguientes recursos:

- PozoWinds - Spots Windsurf Gran Canaria
- Google Maps - Spots Gran Canaria

Dicho centro es uno de los colaboradores del evento anual organizado por la Asociación Profesional de Windsurf (PWA) en Pozo Izquierdo, Gran Canaria. Además, la validez de la información proporcionada en los enlaces mencionados ha sido corroborada por windsurfistas locales.

4. Conclusiones Fase I

Tras estos primeros apartados, se da por concluida la Fase I del proyecto de la que se extraen las siguientes conclusiones:

1. Selección de fuentes de datos confiables:
La elección de fuentes de datos confiables y validadas, como AEMET OpenData y Puertos del Estado, ha sido fundamental para garantizar la calidad y precisión de los datos meteorológicos utilizados en el proyecto. Estos datos son esenciales para el desarrollo de algoritmos predictivos precisos.

2. Validación de información local:
La validación de la información proporcionada por centros deportivos locales y windsurfistas ha permitido identificar los spots más relevantes y adecuados para el estudio. Esta validación local asegura que las recomendaciones de equipo generadas por WindCaddy sean aplicables y útiles para la comunidad de windsurf en Gran Canaria.

3. Metodología de desarrollo estructurada:
La adopción de una metodología de desarrollo estructurada, basada en fases claramente definidas, ha facilitado la gestión del proyecto y la asignación de roles y responsabilidades. Esto ha permitido avanzar de manera eficiente a través de las distintas etapas del proyecto y cumplir con los plazos establecidos.

4. Gestión proactiva de riesgos:
La identificación temprana de riesgos y la implementación de planes de contingencia han sido cruciales para mitigar posibles contratiempos y asegurar la continuidad y el éxito del proyecto. Esto demuestra un enfoque proactivo hacia la gestión de riesgos y la resiliencia ante posibles problemas.

5. Proceso ETL

Las operaciones de extracción, transformación y carga de datos (`ETL`), sobre los orígenes de datos del punto anterior se realizan mediante código `Python`. Básicamente se dividen en dos procesos ETL sobre dos orígenes seleccionados.

Este proceso constituye la primera parte de la Fase I del proyecto, dentro de las 4 en las que se ha subdivido esta.

Figura 4: Flujo de trabajo Fase II [ETL]

5.1 Parametrización del software

La parametrización del software en Python se encuentra en el archivo `constant.py`, el cual es leído por varios módulos del proyecto. Este archivo está dividido en secciones y puede ser configurado según sea necesario para activar o desactivar funciones del software, como se muestra a continuación:

```
# Constants file
# SETUP APP
DO_ETL = True
DO_EXPLORATORY_ANALYSIS = False
DO_PCA = False
DO_TESTMODELS = False
DO_TUNNING = False
DO_VALIDATION = False
```

Este fichero también contiene la parametrización de la base de datos de destino para la carga de las tablas correspondientes bajo el siguiente apartado:

```
# DOCKER LOCAL
DB_NAME = "windcaddy"
DB_USER = "windcaddy"
DB_PASSWORD = "windcaddy"
DB_HOST = "localhost"
DB_PORT = "5432"
```

Existen otras parametrizaciones alojadas en este fichero que se comentarán más adelante y que configuran parámetros adicionales de conexión a fuentes de datos u otros aspectos del proyecto como los modelos comparados.

5.2 Modelo de datos y carga inicial

La carga de los diferentes orígenes de datos residirá en un base de datos `postgreSQL` que persistirá estos para ofrecerlos en la plataforma de etiquetado.

La creación de este modelo reside el fichero `SQL/scheme.sql` donde se crean las tablas necesarias para alojar las descargas de datos desde la AEMET y puertos.es, así como la creación y carga de otras tablas auxiliares.

Tabla	Descripción
AEMET_ESTACION	Tabla maestra de las estaciones de la AEMET
AEMET_HISTORICO	Tabla con los datos históricos descargados desde la AEMET
PUERTOS_PUNTO_MODELO	Tabla maestra de los puntos del modelo predictivo desde puertos.es
PUERTOS_HISTORICO	Tabla con el histórico de datos de los puntos modelo de la AEMET
WC_DEPORTE	Tabla maestra de deportes
WC_MODALIDAD	Tabla maestra de modalidades
WC_SPOT	Tabla maestra con los *spots*
WC_SPOT_WC_MODALIDAD	Tabla que relaciona *spots* con la modalidad del deporte que se practica
WC_SPOT_AEMET_ESTACION	Tabla que relaciona el *spot* con la estación más cercana de la AEMET.
WC_SPOT_PUERTOS_PUNTO_MODELO	Tabla que relaciona el *spot* con el punto de modelo de puertos.es más cercano.

Las tablas subrayadas indican que sólo se crean en este fichero y que su carga se realiza mediante los datos descargados desde las fuentes mencionadas anteriormente.

5.3 Carga desde OpenData AEMET

Desde este repositorio se obtendrán los descriptivos de las estaciones y los históricos de sus registros desde 2013 mediante el código fuente de los siguientes ficheros:

- `ETL/getEstaciones.py`
- `ETL/getHistoricoDiario.py`

Este software se conecta a los servicios `REST` proporcionados por la AEMET, para descargar los datos solicitados y los persiste en una base de datos `postgreSQL` (tablas `AEMET_ESTACION` y `AEMET_HISTORICO`) y en el directorio `DATA/AEMET/JSON`.

La parametrización para esta descarga reside en el fichero `constants.py` del que se muestra una captura a continuación:

```
# AEMET
API_KEY = "Valor API KEY"
ESTACIONES = ["C619I", "C629X", "C649I", "C659H", "C659M", "C689E"]
SERVER_URL = "https://opendata.aemet.es/opendata/api"
TIME_BETWEEN_STATIONS_REQUEST = 10
TIME_BETWEEN_YEARS_REQUEST = 4
```

En esta sección del fichero se parametriza con la `API_KEY` recibida para la descarga de datos desde la AEMET, los indicativos de las estaciones seleccionados para el estudio, la URL base donde se oferta la `API` y dos parámetros que indican la pausa a realizar entre las solicitudes de estaciones y los años para evitar el código `HTTP/429 – Too Many Request`.

5.4 Carga desde Puertos.es

Como se ha comentado anteriormente la descarga desde este origen es solicitada a través de un aplicativo y descargada mediante enlaces.

Así que teniendo este en mente, la descarga de ficheros es manual alojando la colección de ficheros CSV en el directorio `DATA/PUERTOS/CSV`.

El fichero `ETL/getPortsData.py` realiza la carga de estos en la tabla `PUERTOS_HISTORICO` de la base de datos.

5.5 Conjunto de datos a etiquetar

Una vez realizada la carga inicial y la carga particular para cada uno de los orígenes de datos externos, el siguiente paso es fusionar todos los datos para conseguir un conjunto de datos único.

Este proceso se realiza mediante la ejecución del fichero `SQL/create_wc_labeldataset.sql` en la base de datos donde se realiza una consulta entre las diferentes tablas cargadas para obtener la tabla `wc_labeldataset`. La consulta pretende obtener un registro por deporte y modalidad de cada uno de los registros meteorológicos obtenidos del cruce entre los datos de la AEMET y puertos.es

En esta consulta de carga se realiza la conversión de unidades de `m/s` a `nudos` que es la magnitud más popular entre los deportistas. También se descartan algunos registros que se consideran incoherentes, como los que la racha de

viento seas mayor que la velocidad del viento, así como los registros que no se encuentren entre las 07:00h y las 20:00h.

Este proceso se realiza mediante la ejecución en la base de datos del fichero `SQL/create_wc_labeldataset.sql`. Este archivo realiza una consulta entre varias tablas cargadas para generar la tabla `wc_labeldataset`. El propósito de esta consulta es obtener un único registro por cada combinación de deporte y modalidad, a partir de los datos meteorológicos obtenidos del cruce entre la información de AEMET y puertos.es.

Durante esta operación de carga, se efectúa la conversión de unidades de metros por segundo a nudos, la cual es la magnitud más popular entre los deportistas. Además, se descartan registros considerados incoherentes, como aquellos en los que la velocidad de la racha de viento sea menor que la velocidad del viento promedio. También se excluyen los registros que no se encuentren en el intervalo horario de 07:00 a 20:00 horas, ya que se considera residual la práctica de deportes de este tipo en ese horario debido a la falta de luz.

La consulta `SQL` que alimenta esta tabla puede consultarse en el anexo 21.1 de la presente memoria.

6. Etiquetado de los datos

Para obtener las variables objetivo se ha desarrollado y puesto a disposición de los *riders* que quieran participar en el estudio, un formulario web mediante el cual se etiquetarán los registros.

- http://windcaddy.hopto.org

Este proceso de etiquetado es el segundo paso dentro del flujo de la Fase I del proyecto.

Figura 5: Flujo de trabajo Fase II [ETIQUETADO]

6.1 Tecnologías usadas y arquitectura

Este formulario está diseñado con dos partes fundamentales: un *frontend* desarrollado en `React`, desplegado mediante contenedores `Docker` que usan como servidor web `Nginx`; y un *backend* que emplea `Spring Boot`. El *backend* se encarga de todas las operaciones necesarias para la persistencia y consulta de datos, en una base de datos `PostgreSQL` también desplegada mediante `Docker`. El código fuente de este formulario se adjunta en la entrega de este proyecto.

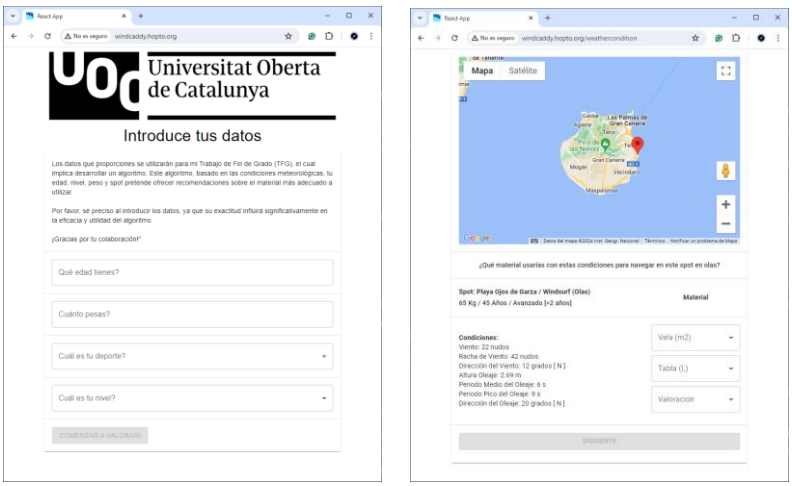


Figura 6: Capturas del formulario de etiquetado.

6.2 Promoción

Para la promoción de formulario se han realizado campañas de promoción en redes sociales y centros locales mediante la publicación de los siguientes reclamos:

Figura 7: Cartelería de promoción formulario web

7. Análisis exploratorio

El objetivo analítico del proyecto es explorar y comprender los patrones y relaciones presentes en los datos meteorológicos y de uso del equipo por parte de los deportistas acuáticos mediante técnicas avanzadas de análisis de datos.

Este propósito implica identificar correlaciones entre las condiciones meteorológicas y las elecciones de equipo, así como comprender las preferencias y necesidades individuales de los deportistas.

A través de este análisis, WindCaddy busca obtener información clave que alimente el desarrollo del algoritmo de recomendación de equipo, permitiendo así ofrecer pronósticos más precisos y sugerencias personalizadas que mejoren el rendimiento deportivo y la seguridad de los usuarios.

El análisis exploratorio es el tercer paso dentro del flujo de trabajo para esta primera fase.

Figura 8: Flujo de trabajo Fase II [ANALISIS EXPLORATORIO]

7.1 Exploración del conjunto de datos

El conjunto de datos obtenido tras el etiquetado consta de 33 variables que se describen en el siguiente cuadro:

Variable	Tipo de dato	Descripción
id	int64	Identificador registro
id_spot	int64	Identificador spot
spot	object	Nombre spot
fecha	date	Fecha del día (AAAA-MM-DD), convertida a TIMESTAMP
dir	object	Dirección de la racha máxima en decenas de grado (99 = dirección variable) (88 = sin dato)
velmedia	float64	Velocidad media del viento en nudos
racha	float64	Racha máxima del viento en nudos
altura_oleaje	float64	Altura Signif. del Oleaje(m)
periodo_medio_oleaje	float64	Periodo Medio(s)
periodo_pico_oleaje	float64	Periodo Pico(s)
direccion_oleaje	float64	Direcc. Media de Proced.(0=N,90=E)

velocidad_viento	float64	Velocidad del viento en nudos
direccion_viento	float64	Direc. de proced. del Viento(0=N,90=E)
velocidad_corriente	float64	Velocidad corriente(m/s)
direccion_corriente	float64	Dir. de prop. de la Corriente (0=N,90=E)
id_deporte	int64	Identificador deporte
deporte	object	Descripcion deporte
id_modalidad	int64	Identificador modalidad
modalidad	object	Descripción modalidad
valoracion	float64	Valoración de las condiciones
perfil_rider	object	Perfil rider (Novel, Avanzado, Pro)
peso_rider	float64	Peso del rider en Kg
kite_size	float64	Tamaño de la cometa para Kitesurfing m^2
sail_size	float64	Tamaño de la vela para Windsurfing m^2
wing_size	float64	Tamaño de la vela de Wingsurfing m^2
kite_board_size	float64	Tamaño de la tabla de Kitesurf en cms
kite_board_type	object	Tipo de la tabla de Kitesurf (Twintip, Surfboard)
wind_board_size	float64	Tamaño de la tabla de Windsurf en litros
wind_board_type	object	Tipo de la tabla de Windsurf (Olas, Slalom, Freestyle, Freeride)
wing_board_size	float64	Tamaño de la tabla de Wingsurf en litros
wing_ffoil_size	float64	Tamaño ala frontal foil (Area proyectada cm^2)
labeled	boolean	Registro etiquetado
edad_rider	float64	Edad en años del rider

7.1.1 Hechos a estudiar

- **Wind Board Size (Tamaño de la tabla de windsurf)**: Este es uno de los objetivos principales del estudio y representa el tamaño de la tabla utilizada en windsurfing. Podría estar influenciado por diversas condiciones meteorológicas y características del rider.

- **Sail Size (Tamaño de la vela para windsurfing)**: Otra variable objetivo del estudio que representa el tamaño de la vela utilizada en windsurfing. Al igual que el tamaño de la tabla, puede verse afectada por diversas condiciones meteorológicas y características del rider.

- **Valoración de las Condiciones**: Esta es una medida subjetiva de las condiciones, proporcionada por los riders. Puede depender de una combinación de factores meteorológicos y preferencias personales del rider según su perfil.

7.1.2 Dimensiones Meteorológicas:

- **Velocidad Media del Viento (velocidad_viento)**: La velocidad media del viento en nudos puede afectar la experiencia de navegación en deportes como el windsurfing. Un viento más fuerte podría requerir un equipo más pequeño para controlar.

- **Racha Máxima del Viento (racha)**: Las ráfagas de viento pueden ser importantes para determinar la capacidad de controlar la vela y la tabla.

- **Altura Significativa del Oleaje (altura_oleaje)**: La altura del oleaje puede influir en la navegación y el rendimiento del rider en deportes como el windsurfing. Oleajes más grandes podrían requerir tamaños de equipo diferentes.

- **Dirección Media de Procedencia del Viento (direccion_viento)**: La dirección del viento puede ser crítica para determinar el recorrido y la maniobrabilidad en deportes acuáticos.

- **Velocidad de la Corriente (velocidad_corriente)**: La corriente puede afectar la velocidad y dirección del movimiento de la tabla y la vela en deportes como el windsurfing.

- **Dirección de la Corriente (direccion_corriente)**: La dirección de la corriente puede afectar la ruta y el rendimiento del rider.

- **Dirección de la racha (dir):** La dirección de la racha con respecto a la dirección media puede influir en la experiencia de navegación debido a las correcciones inmediatas a realizar en la posición para adaptarse a esta.

7.1.3 Dimensiones Temporales:

- **Fecha (fecha)**: La fecha puede ser importante para el análisis temporal, ya que las condiciones meteorológicas pueden variar según la temporada o el mes del año.

7.1.4 Dimensiones relativas al *rider*:

- **Nivel (perfil_rider)**: Indica la experiencia que tiene el *rider* en la práctica del deporte que puede ser determinante a la hora de la valoración de las condiciones o aparejo a usar.

- **Peso (peso_rider)**: El peso del usuario es una de las principales razones para la selección del tamaño de vela y tabla ya que, a mayor peso mayor necesidad de tamaño de vela y tabla.

- **Edad (edad_rider)**: Se intuye que esta variable, aunque menos determinante que las anteriores, podría ser relevante a la hora de la valoración de las condiciones.

7.1.5 Exclusiones

Se procede a excluir las siguientes variables:

Variable	Tipo de dato	Motivo exclusión
id	int64	No aporta valor al estudio
id_spot	int64	No aporta valor al estudio
velmedia	float64	No se ha utilizado esta variable para la selección del aparejo y la valoración en el formulario de etiquetado, en favor de la variable "velocidad_viento"
labeled	Bool	Indica que el registro ha sido etiquetado, no relevante.

7.1.6 Transformaciones de datos

- Se procede a actualizar los registros de la columna racha a `None`, cuando estos tengan un valor 88, por la definición de estos.
- Se procede a cambiar el tipo de datos de la columna `dir` de `object` a `int64`.
- Se procede a añadir el punto cardinal equivalente a la dirección de la racha máxima viento en grados de acuerdo a la tabla siguiente, previa transformación de tipo de datos (object → int64) y de decenas de grados a grados:

Variable	Valor	Variable	Valor
dir	338 a 44	dir_card	N
dir	45 a 68	dir_card	NE
dir	69 a 112	dir_card	E
dir	113 a 157	dir_card	SE
dir	158 a 202	dir_card	S
dir	203 a 247	dir_card	SO
dir	248 a 292	dir_card	O
dir	293 a 337	dir_card	NO
dir	990	dir_car	VAR

- Se procede a añadir el punto cardinal equivalente a la dirección del viento en grados de acuerdo a:

Variable	Valor	Variable	Valor
direccion_viento	338 a 44	dirección_viento_card	N
direccion_viento	45 a 68	dirección_viento_card	NE
direccion_viento	69 a 112	dirección_viento_card	E
direccion_viento	113 a 157	dirección_viento_card	SE
direccion_viento	158 a 202	dirección_viento_card	S
direccion_viento	203 a 247	dirección_viento_card	SO
direccion_viento	248 a 292	dirección_viento_card	O

direccion_viento	293 a 337	dirección_viento_card	NO

- Se procede a cambiar el tipo de datos del campo `fecha` de `object` a `datetime`.

7.1.7 Codificaciones de datos

- Se procede a añadir una columna numérica id_perfil_rider de acuerdo a la siguiente tabla:

Variable	Valor	Variable	Valor
perfil_rider	'Novel [<2 años]'	Id_perfil_rider	1
perfil_rider	'Avanzado [>2 años]'	*Id_perfil_rider*	2
perfil_rider	'Pro'	Id_perfil_rider	3

7.1.8 Discretizaciones de datos

- Se agrega una nueva columna `mes` que tendrá el nombre del mes con 3 letras.

- Se agrega una nueva columna `año` que tendrá el valor del año del registro.

7.2 Preprocesamiento y gestión de características

7.2.1 Valores nulos

Se gestionará cada característica como se define a continuación:

- Porcentaje de nulos > 95%, se descarta la característica
- Porcentaje de nulos > 50% y <95% no se actúa sobre ellos.
- Porcentaje de nulos < 50%, se estudiará cada característica y se aplicará alguna de las siguientes:
  - Sustituirlos por la media estadística
  - Sustituirlos por la mediana estadística
  - Eliminar los registros afectados.

Tras la aplicación del tratamiento de nulos se obtiene la siguiente tabla, donde se ha reducido la dimensionalidad en 7 variables.

Variable	Nulos %	Eliminada
spot	0.00	No
fecha	0.00	No
dir	0.00	No
racha	0.00	No
altura_oleaje	0.00	No
periodo_medio_oleaje	0.00	No
periodo_pico_oleaje	0.00	No

direccion_oleaje	0.00	No
velocidad_viento	0.00	No
direccion_viento	0.00	No
velocidad_corriente	57.34	No
direccion_corriente	57.34	No
id_deporte	0.00	No
deporte	0.00	No
id_modalidad	0.00	No
modalidad	0.00	No
valoracion	0.00	No
perfil_rider	0.00	No
peso_rider	0.00	No
kite_size	**100.00**	**Sí**
sail_size	60.88	No
wing_size	**100.00**	**Sí**
kite_board_size	**100.00**	**Sí**
kite_board_type	**100.00**	**Sí**
wind_board_size	60.88	No
wind_board_type	**100.00**	**Sí**
wing_board_size	**100.00**	**Sí**
wing_ffoil_size	**100.00**	**Sí**
labeled	0.00	No
edad_rider	0.00	No
id_perfil_rider	0.00	No

Nótese que esta reducción obedece a que no son variables ofertadas para esta primera aproximación del algoritmo donde se ha definido el alcance a Windsurfing en la modalidad de olas y que por tanto el formulario de etiquetado no oferta ningún dato referente a otros deportes y modalidades.

7.2.2 Varianza estadística

Se descartarán las características numéricas con una varianza próxima a cero (por debajo de 0,01), ya que carecen de variabilidad y no describen ninguna propiedad de los registros del conjunto de datos.

Tras la ejecución del tratamiento del conjunto de datos por la varianza estadística de sus componentes se obtiene la siguiente tabla:

Variable	Varianza	Eliminada
racha	78.66	No
altura_oleaje	0.37	No
periodo_medio_oleaje	2.00	No
periodo_pico_oleaje	9.70	No
direccion_oleaje	7853.77	No
velocidad_viento	48.44	No

direccion_viento	13165.45	No
velocidad_corriente	0.04	No
direccion_corriente	10049.36	No
id_deporte	**0.00**	**Sí**
id_modalidad	**0.00**	**Sí**
valoracion	1.13	No
peso_rider	92.55	No
sail_size	0.51	No
wind_board_size	132.56	No
edad_rider	216.54	No
id_perfil_rider	0.54	No

Nótese que estas variables han sido eliminadas debido al alcance actual del estudio mencionado en el apartado anterior.

7.2.3 Revisión clases objetivo

En el presente subapartado se estudia la sobrerrepresentación (*oversampling*) y la infrarrepresentación (*undersampling*) de las clases objetivo (`sail_size`, `wind_board_size`, `valoración`) en el conjunto de datos.

El siguiente cuadro da una visión preliminar de las clases objetivo dentro del conjunto de datos.

	sail_size	wind_board_size	valoracion
count	2979	2979	7616
mean	6.0	91	2.0
std	1.0	12	1.0
min	3.0	65	1.0
25%	5.0	80	1.0
50%	6.0	90	1.0
75%	6.0	100	2.0
max	6.0	120	5.0

Siendo Gran Canaria el *spot* la media del tamaño de vela y tabla en el conjunto de datos sugiere que las condiciones ofertadas por el formulario web a valorar han sido bastante por debajo de las condiciones que se pueden considerar normales en la isla para la práctica de windsurf (viento > 20 nudos). Esta intuición se corrobora con la media de las valoraciones que es bastante baja.

A continuación, se presentan las proporciones en las diferentes variables objetivos:

wind_board_size	Proporción (%)
90.0	15.01
80.0	14.74
95.0	13.56
100.0	13.19
85.0	12.89
105.0	10.84
75.0	7.52
110.0	5.07
70.0	4.20
115.0	1.81
120.0	0.84
65.0	0.34

sail_size	Proporción (%)
6.5	31.62
6.0	20.85
5.5	20.44
5.0	13.39
4.5	6.58
4.7	4.67
4.2	1.11
4.0	0.84
3.5	0.20
3.7	0.20
3.3	0.10

Valoración	Proporción (%)
1.0	60.98
2.0	22.77
3.0	8.32
5.0	4.44
4.0	3.49

Existe un desequilibrio en las clases `sail_size` y `valoracion` ya que hay una clara disparidad en las proporciones entre las diferentes categorías en ambas variables, indicando un desequilibrio de clases en estas.

7.2.4 Balanceo de clases objetivos

Se realizará un balanceo de clases para las variables objetivo mencionadas en el apartado anterior con el objetivo de minimizar el riesgo de sesgo del algoritmo hacia alguna de las clases más frecuentes y no generalice correctamente para las clases minoritarias.

Para ello se utilizará la técnica de *oversampling* a `valoración` y `sail_size` mediante el uso de la biblioteca `resample` de `sklearn.utils`.

Este es el resultado final en las variables objetivos:

wind_board_size	Proporción (%)
75	22.18
70	16.18
80	14.04
95	10.12
120	8.40
85	7.18
90	5.30
100	5.03
65	4.29
110	2.89
115	2.24
105	2.14

sail_size	Proporción (%)
5.0	14.66
6.5	12.07
5.5	10.79
3.7	9.45
6.0	8.43
4.5	7.78
4.7	7.65
4.0	7.53
3.3	7.40
3.5	7.37
4.2	6.87

Valoración	Proporción (%)
2	20.00
3	20.00
5	20.00
4	20.00
1	20.00

7.2.5 Visión general de las variables objetivo

En el cuadro siguiente se presenta una comparación resumida de la información estadística de los datos antes y después de realizar el balanceo de clases, centrándose en las variables objetivo del estudio.

	sail_size		wind_board_size		valoracion	
	pre	post	pre	post	pre	post
count	2979	16635	2979	16635	7616	16635
mean	6.0	5.0	91	86	2.0	3.0
std	1.0	1.0	12	16	1.0	1.0
min	3.0	3.0	65	65	1.0	1.0
25%	5.0	4.0	80	75	1.0	2.0
50%	6.0	5.0	90	80	1.0	3.0
75%	6.0	6.0	100	95	2.0	4.0
max	6.0	6.0	120	120	5.0	5.0

El tamaño promedio de la vela era de 6.0 con una variabilidad baja, indicada por una desviación estándar de 1.0. La mayoría de las velas se encontraban en el rango de 5.0 a 6.0. Tras el balanceo, el tamaño promedio disminuyó a 5.0,

manteniendo una variabilidad similar con una desviación estándar de 1.0. El rango intercuartil se desplazó ligeramente hacia tamaños más pequeños, entre 4.0 y 6.0.

El tamaño promedio de la tabla era de 91 con una desviación estándar de 12, lo que indica una variabilidad moderada. La mayoría de las tablas se ubicaban en el rango de 80 a 100. Después del balanceo, el tamaño promedio se redujo a 86, y la variabilidad aumentó con una desviación estándar de 16. El rango intercuartil se desplazó hacia tamaños más pequeños, oscilando entre 75 y 95.

La valoración promedio era de 2.0 con una variabilidad baja, mostrada por una desviación estándar de 1.0. La mayoría de las valoraciones estaban en el rango de 1.0 a 2.0. Tras el balanceo, la valoración promedio aumentó a 3.0, manteniendo una variabilidad constante con una desviación estándar de 1.0. El rango intercuartil se expandió ligeramente, moviéndose entre 2.0 y 4.0.

Se concluye que el balanceo de clases resultó en cambios menores en las medidas centrales de las variables, pero manteniendo en su mayoría la variabilidad en torno a los valores previos.

7.2.6 Estudio de frecuencias de variables

En las siguientes gráficas, se realiza un análisis detallado de las frecuencias de valores presentes en el conjunto de datos. Estas visualizaciones nos permiten examinar la distribución y la ocurrencia de cada valor, proporcionando comprensión profunda muy valiosa sobre la variabilidad y la representatividad de los datos en distintas variables.

Este análisis es fundamental para comprender mejor la naturaleza y la estructura del conjunto de datos, así como para identificar posibles patrones, anomalías o áreas de interés que puedan requerir una atención específica en el proceso de análisis posterior.

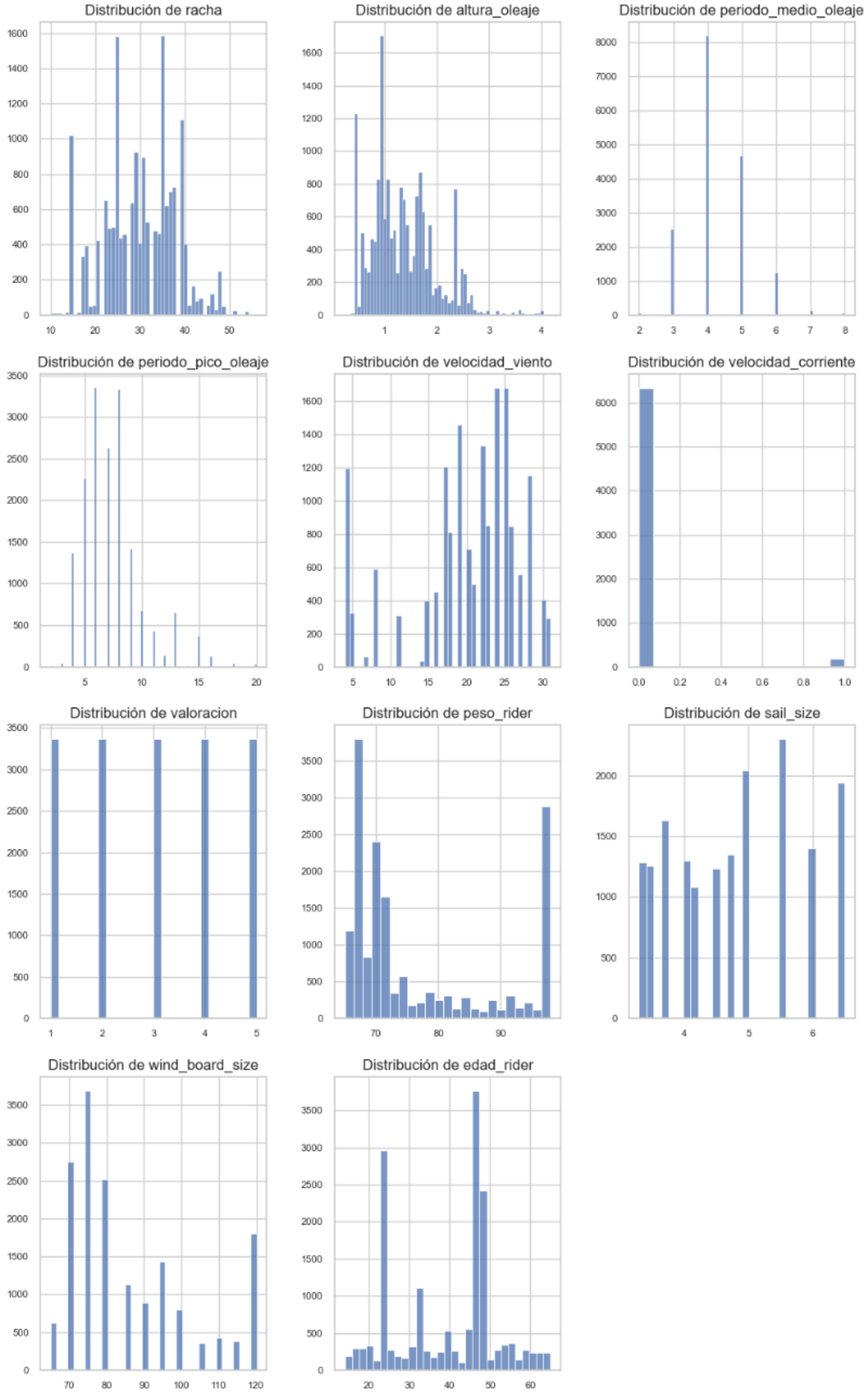


Figura 9: Frecuencias de las dimensiones numéricas

Racha:
Las rachas varían en intensidad, con valores que van desde 15.0 hasta 39.0. Las rachas más comunes tienen valores alrededor de 35.0 y 25.0, una racha con varios nudos de diferencia con respecto a la velocidad media del viento, puede ser influir negativamente en la experiencia de navegación.

Altura del oleaje:
La altura del oleaje muestra una variedad de valores, desde 0.44 hasta 2.35 metros. Las alturas más comunes son 0.44 y 0.95 metros, son alturas que dependiendo del periodo son accesibles a todos los niveles de *riders.*

Periodo medio del oleaje:
Los periodos medios del oleaje se distribuyen principalmente entre 3.0 y 6.0, siendo 4.0 el más común. Los periodos superiores a 8 en conjunción de la altura del oleaje podrían contribuir a que las condiciones no fuesen aptas para todos los niveles.

Periodo pico del oleaje:
Los periodos pico del oleaje varían entre 4.0 y 15.0, siendo 6.0 y 8.0 los más comunes.

Velocidad de la corriente: La mayoría de las observaciones muestran una velocidad de corriente de 0.0, con solo 187 observaciones con velocidad de corriente de 1.0. Esta variable no suele ser tomada en cuenta por los deportistas a la hora de seleccionar el aparejo, aunque puede ser importante en condiciones de viento mediocres.

Valoración: La valoración se distribuye uniformemente entre 1.0 y 5.0, debido al proceso de *oversampling* realizado a la muestra.

Peso del rider: Los pesos de los riders varían, con 67.0 Kg como el peso más común seguido de 98.0 Kg. Se intuye que el peso guarda una relación lineal con el tamaño de tabla y vela a seleccionar.

Tamaño de la vela: Los tamaños de vela varían, con 5.5 como el más común, seguido de 5.0. Estos datos sugieren que la mayoría de las valoraciones que se han recogido probablemente sean de spots que no se encuentran entre los más populares, ya que el conjunto de velas que más se usa en la isla va desde 3.7 a 4.7.

Volumen tabla de windsurf:
Los tamaños de tabla de windsurf varían, siendo las más comunes las que cuentan con un volumen entre 70 y 80 litros.

Edad del *rider*:
Las edades de los *riders* están distribuidas de manera diversa, con 46.0 como la más común.

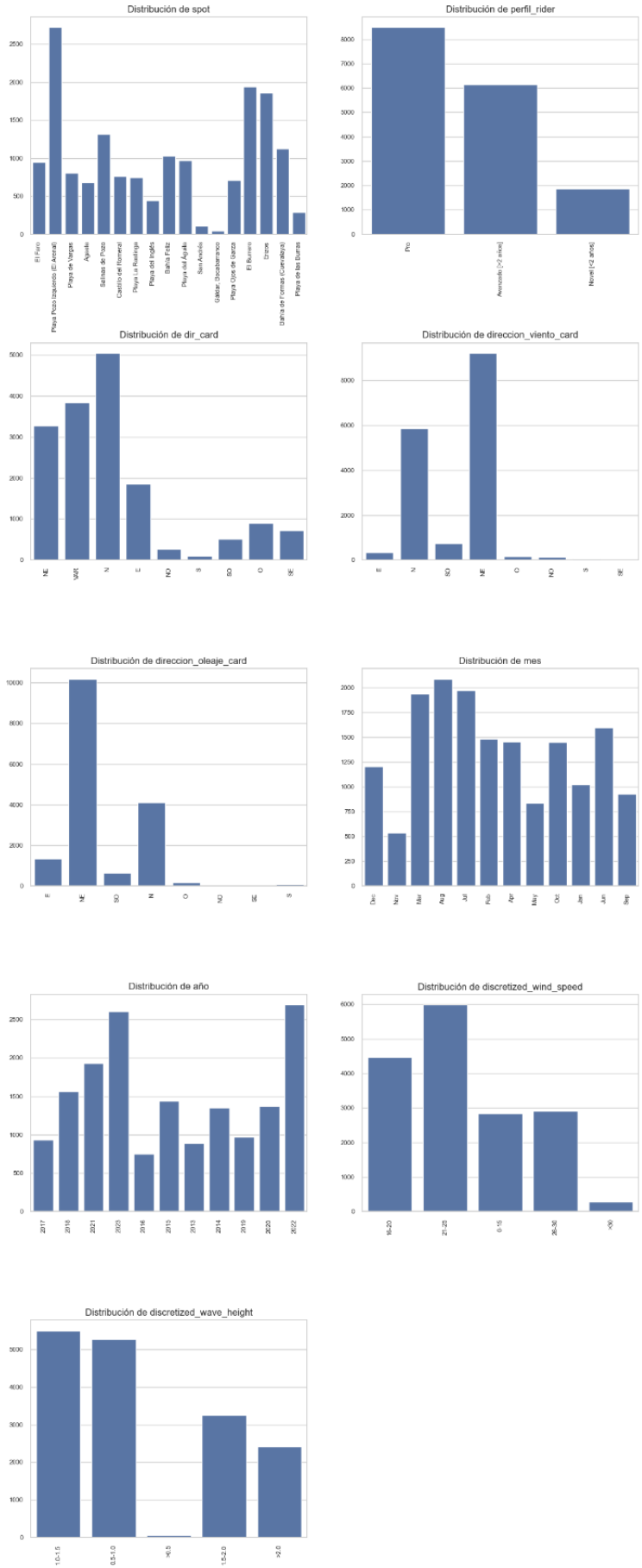


Figura 10: Frecuencias de las dimensiones no numéricas

- **Spot:**
  La playa Pozo Izquierdo (El Arenal) y Bahía Feliz, tienen el mayor número de ocurrencias en el conjunto de datos etiquetados.

- **Perfil del *rider*:**
  Esta distribución revela la composición de los *riders* según su nivel de habilidad. La mayoría de los *riders* se clasifican como "`Pro`", seguidos por aquellos con un nivel de habilidad "`Avanzado [>2 años]`", mientras que los *riders* novatos representan la menor proporción en el conjunto de datos.

- **Dirección de la racha:**
  Esta distribución muestra la dirección la racha de viento registrada, representada en términos de puntos cardinales. La dirección predominante de la racha parece ser del norte (N) y variante (VAR), lo que indica una variedad en las condiciones de viento registradas.

- **Dirección del viento:**
  Muestra la dirección del viento, siendo la predominante la componente noreste (NE), seguida por norte (N), lo que sugiere una tendencia consistente en la dirección del viento registrada. Son famosos los “Alisios” que azotan la isla en esas direcciones.

- **Dirección del oleaje:**
  Muestra la dirección del oleaje y sugiere que las direcciones predominantes del oleaje son NE, N y E, mientras que las direcciones menos comunes son SE, NO, S, O y SO.

- **Mes:**
  Muestra la distribución registros según el mes del año. La distribución es bastante uniforme, con una mayor cantidad de eventos registrados en los meses de verano (Jun, Jul, Aug).

- **Año:**
  Muestra la distribución de registros según el año. Parece haber una distribución variada a lo largo de los años, con más registros en los años recientes, como 2022 y 2021, lo que podría indicar un aumento en la recopilación de datos con el tiempo.

- **Velocidad del viento:**
  Esta distribución clasifica la velocidad del viento en rangos discretos. La mayoría de las observaciones registran velocidades de viento entre 21 y 25 nudos, seguidas por velocidades entre 16 y 20 nudos, lo que sugiere condiciones de viento Fresco/Fuerte según Beaufort.

- **Altura oleaje:**
  Similar a la distribución de velocidad del viento, pero se enfoca en la altura de las olas. La mayoría de las observaciones registran alturas de olas

entre 0.5 y 1.0 metros, seguidas por alturas entre 1.0 y 1.5 metros, indicando condiciones de oleaje moderado.

7.2.7 Correlación entre las variables

En este subapartado se estudia la correlación entre las dimensiones continuas del conjunto de datos obtenido, para ello se apoya en una matriz de correlación.

La matriz de correlación proporciona información sobre la relación lineal entre pares de variables en un conjunto de datos. Las correlaciones varían entre -1 y 1, donde:

- 1 indica una correlación positiva perfecta,
- -1 indica una correlación negativa perfecta
- 0 indica falta de correlación.

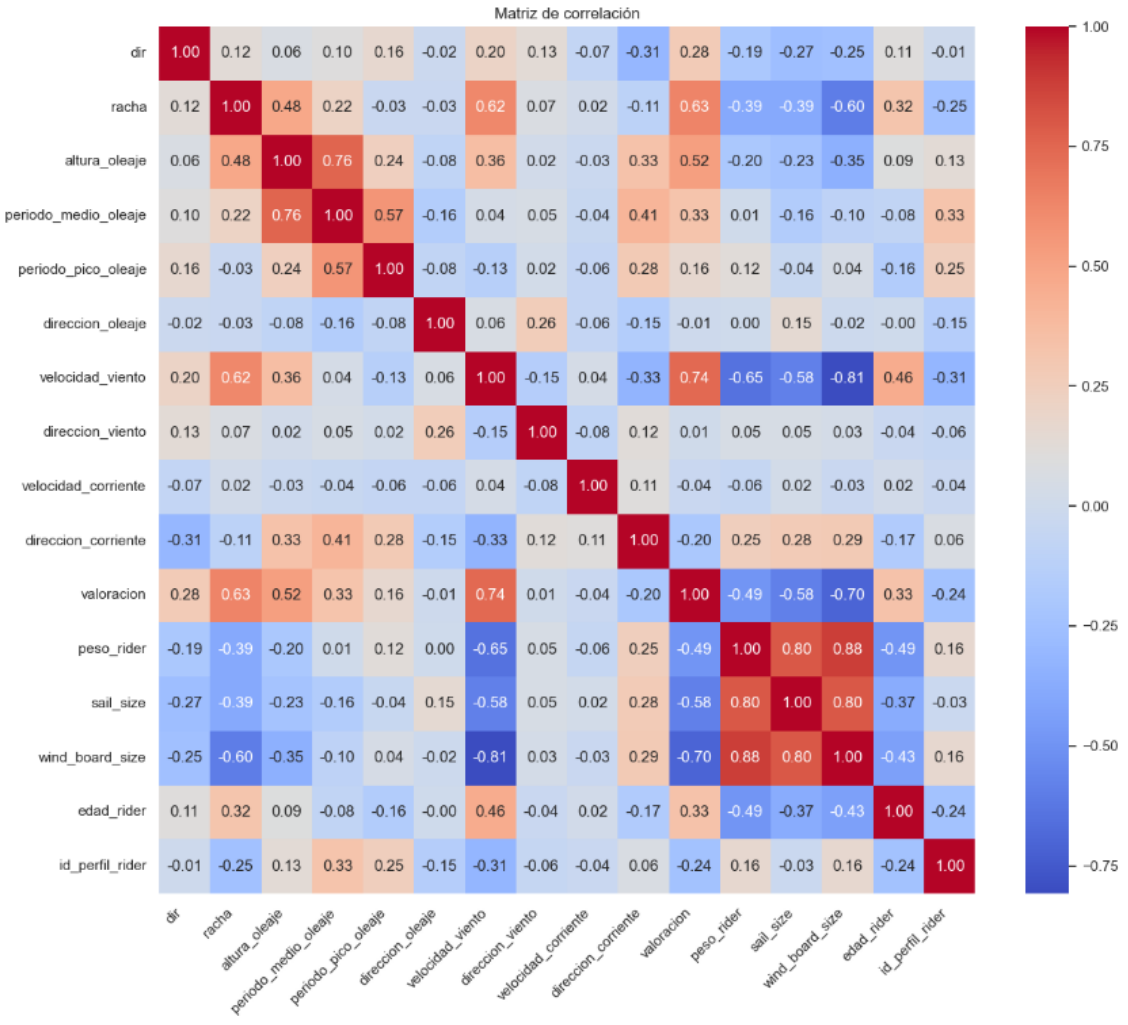


Figura 11: Matriz de correlación Pearson de las variables continuas

Variables con correlación positiva (>0.5):

Estas correlaciones sugieren que estas variables tienden a aumentar o disminuir juntas:

Par de variables		Coef. Correlación
racha	velocidad_viento	0.62
racha	valoracion	0.63
altura_oleaje	periodo_medio_oleaje	0.76
altura_oleaje	valoracion	0.52
periodo_medio_oleaje	periodo_pico_oleaje	0.57
velocidad_viento	valoracion	0.74
peso_rider	sail_size	0.80
peso_rider	wind_board_size	0.88
sail_size	wind_board_size	0.80

Variables con correlación negativa (<-0.5):

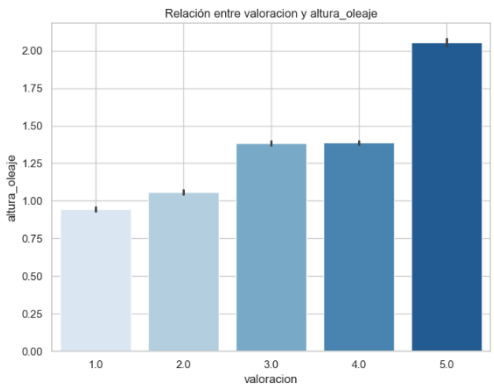
Estas variables tienden a moverse en direcciones opuestas.

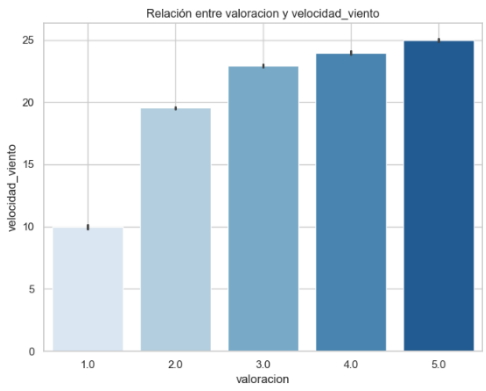
Par de variables		Coef. Correlación
racha	wind_board_size	-0.60
velocidad_viento	peso_rider	-0.65
velocidad_viento	sail_size	-0.58
velocidad_viento	wind_bord_size	-0.81
valoracion	sail_size	-0.58
Valoración	wind_board_size	-0.70

Los pares no listados en los cuadros anteriores se consideran que tienen una baja relación lineal entre sí.

7.2.8 Relación valoración vs variables

En este subapartado se analiza la relación entre ciertas variables del conjunto de datos y la valoración asignada a esas condiciones. El objetivo es obtener una primera impresión sobre las preferencias de los *riders* locales.





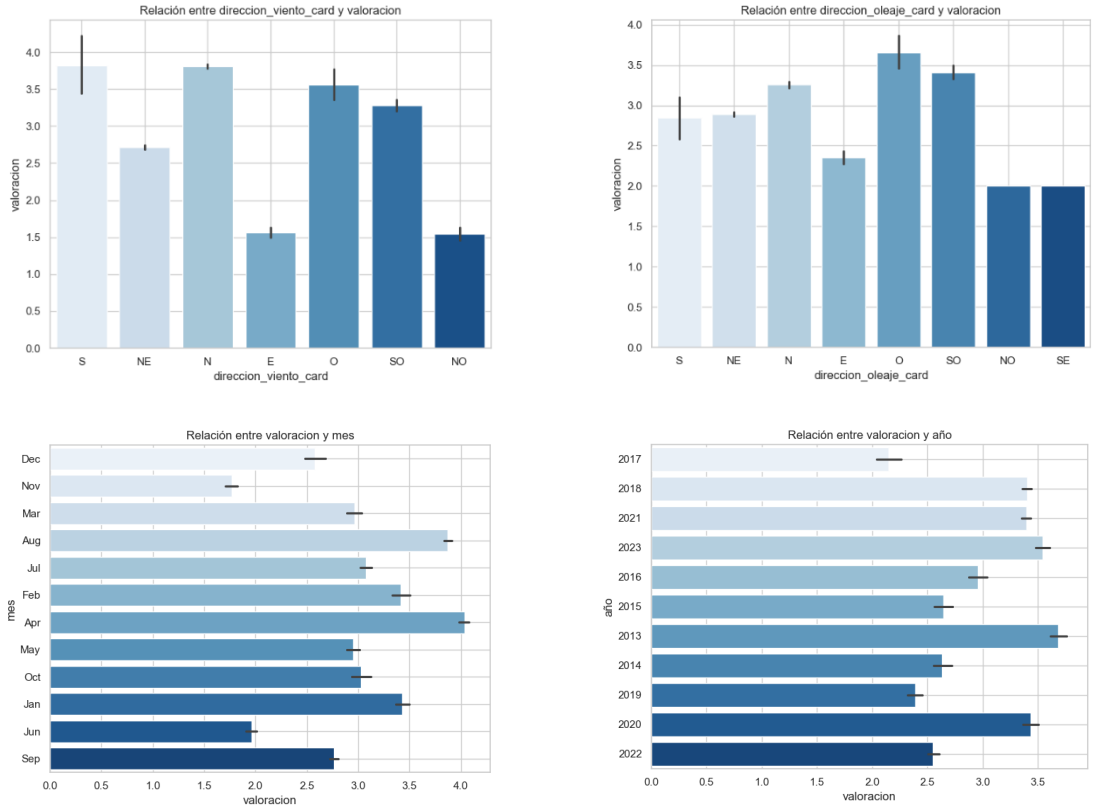


Figura 12: Relación de valoración con algunas variables

Después de analizar los gráficos anteriores, se observa una relación lineal entre la valoración y dos variables clave: la altura de la ola y la velocidad del viento. Estos hallazgos sugieren que, en general, las condiciones son más favorables cuando estas variables tienen valores más altos.

Sin embargo, en lo que respecta a la dirección del viento, se destaca un resultado sorprendente: las valoraciones más altas se atribuyen al viento proveniente del norte y del sur, a pesar de que la preferencia local suele ser para la componente NE. Esta discrepancia plantea interrogantes sobre las condiciones específicas que podrían estar influenciando estas percepciones.

En cuanto a la dirección de las olas, se observa una distribución bastante uniforme en relación con la valoración, con las componentes NO y SE siendo las menos valoradas.

Los últimos dos gráficos analizan variables temporales, donde se destaca una valoración más alta durante las sesiones de verano. Además, se observa un interés particular por los meses de febrero y enero, los cuales son conocidos por tener más oleaje en las islas, aunque el viento no sea tan favorable.

Los gráficos siguientes analizan las tres variables que, a primera vista, suelen ser las más consultadas por los deportistas al decidir el aparejo. Se presentan las valoraciones por perfil del *rider* con el objetivo de discernir qué aspectos valora más cada tipo de perfil.

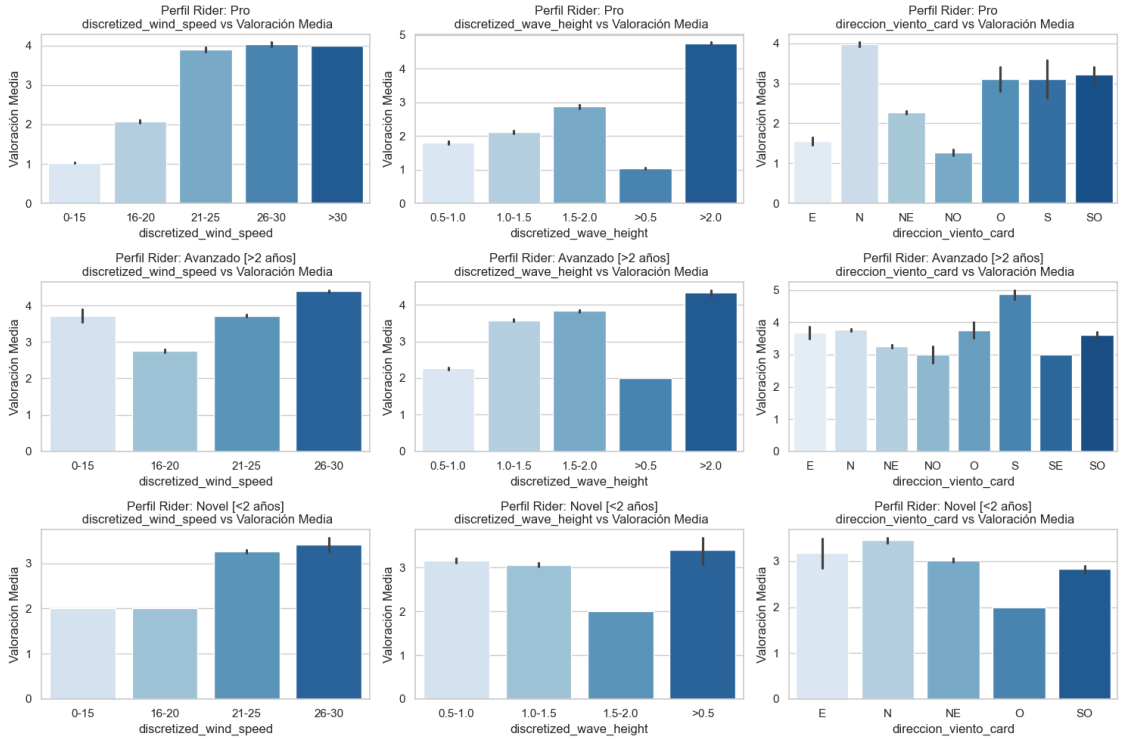


Figura 13: Relación valoración por perfil con algunas variables

No se observa una gran diferencia en la valoración en la velocidad del viento entre los diferentes perfiles, siendo el rango de 26 a 30 nudos el preferido en general.

En lo que respecta a la altura del oleaje, se puede notar una tendencia lineal entre los perfiles más experimentados y la valoración, mientras que los riders novatos parecen preferir condiciones menos exigentes.

Por último, la valoración de la dirección del viento no es uniforme como podría esperarse. Aunque las direcciones N y NE suelen recibir valoraciones positivas en general, es interesante notar que condiciones como O, S y SO, que son comunes en los spots con más oleaje en la isla (costa norte y noroeste), son mejor valoradas por los riders más experimentados. Esto posiblemente se deba a que estas condiciones coinciden con una altura de oleaje más significativa.

7.2.9 Normalización de datos

En la fase de preprocesamiento de datos, se llevan a cabo una serie de transformaciones y ajustes para preparar el conjunto de datos para su análisis y modelado. Uno de los pasos esenciales en este proceso es la normalización de datos.

La normalización es una técnica que ajusta los valores de las características a una escala común, lo que es fundamental para garantizar que las características contribuyan de manera equitativa durante el entrenamiento de modelos de aprendizaje automático.

7.2.9.1 Simplificación del conjunto de datos

Previo a la normalización en sí, se procede a eliminar algunas variables con el objetivo de simplificar y optimizar el conjunto de datos para el análisis. Esto puede conducir a una representación más clara y eficiente de las relaciones y patrones subyacentes en los datos, facilitando así la interpretación de los resultados y mejorando el rendimiento de los modelos de aprendizaje automático subsiguientes.

- Variables de identificación
  [`spot`]

  Esta variable puede ser redundante o no contribuir significativamente al análisis, especialmente si hay suficientes otras variables que describen las condiciones del spot.

- Variables de temporalidad
  [`fecha, mes, año`]

  El algoritmo emitirá una valoración y sugerencia con las condiciones actuales del *spot*, por lo que estas variables no se consideran significativas.

- Variables discretizadas
  [`discretized_wind_speed, discretized_wave_height`]

  Al utilizar PCA, es preferible trabajar con variables continuas para capturar mejor las variaciones en los datos. Las variables discretizadas pueden perder información y reducir la capacidad de PCA para identificar patrones y relaciones significativas en los datos.

- Variables derivadas o transformadas
  [`dir_card, direccion_viento_card, direccion_oleaje_card`]

  Estas variables son transformaciones de otras variables, es más informativo trabajar con las variables originales para evitar pérdidas de información y posibles errores en las transformaciones.

7.2.9.2 Codificaciones one-hot

La codificación one-hot es una técnica comúnmente utilizada para transformar variables categóricas en una representación numérica que puede ser utilizada por algoritmos de aprendizaje automático.

La codificación one-hot es una elección adecuada para la variable `id_perfil_rider` con valores 1, 2 y 3, ya que permite una representación

binaria independiente de cada categoría, preservando la información categórica y mejorando la interpretación y el rendimiento del modelo.

7.2.9.3 Variables continuas

En el proceso de análisis de datos, es común encontrarse con variables que varían significativamente en sus escalas y unidades. Esta variabilidad puede influir negativamente en el rendimiento de ciertos modelos de aprendizaje automático, especialmente aquellos sensibles a las diferencias de escala entre características.

La estandarización es una técnica de normalización que ajusta las variables para que tengan una media de 0 y una desviación estándar de 1. Esto permite que todas las características se expresen en una escala común, facilitando así la comparación y la interpretación de los coeficientes en los modelos.

Las variables `racha`, `altura_oleaje`, `periodo_medio_oleaje`, `periodo_pico_oleaje`, `velocidad_viento`, `velocidad_corriente`, `peso_rider`, `sail_size`, `wind_board_size`, `edad_rider` serán normalizadas mediante esta técnica.

7.2.9.4 Variables circulares

La dirección en grados es una variable circular, lo que significa que los valores están en un rango de 0 a 359 grados. Cuando se trata de normalizar variables circulares, no es adecuado utilizar las técnicas de normalización estándar que se aplican a variables continuas, para estas variables, es más apropiado utilizar la transformación de "circulo-circulo" o "circulo-lineal", que mapea el rango circular a un espacio lineal mientras conserva la estructura circular de los datos. [5] [6] [7]

Una transformación común para este propósito es la transformación de coseno y seno, para este proyecto se usará la transformación de coseno (componente x) de acuerdo a:

$$\cos(grados) = \cos\left(\frac{grados \times 2\pi}{360}\right)$$

Las variables `direccion_viento`, `dir`, `direccion_oleaje` y `direccion_corriente` serán normalizadas con esta técnica.

7.2.9.5 Tratamiento para los valores NaN

La transformación de valores `NaN` (`Not a Number`) a su media es una estrategia comúnmente utilizada para manejar datos faltantes en conjuntos de datos antes de aplicar técnicas de análisis como PCA (Análisis de Componentes Principales).

Al reemplazar los valores faltantes con la media de la columna correspondiente, se conserva la estructura general de los datos y se mitiga el impacto de los valores faltantes en el análisis. Esto permite que el algoritmo PCA opere de manera más eficiente y precisa, ya que los valores faltantes no introducen ruido o distorsiones no deseadas en la estructura de los datos.

7.2.10 Conjunto de datos final

Tras aplicar los procesos de eliminación de variables redundantes, codificación one-hot para variables categóricas y normalización de las variables continuas y circulares seleccionadas, hemos obtenido un conjunto de datos normalizado y optimizado. Este dataset preparado está ahora listo para aplicar el Análisis de Componentes Principales (PCA).

El archivo resultante del proceso, que servirá como entrada para el proceso de PCA, se genera en la siguiente ruta:

- `DATA/PCA/pca_input.json`

8. Análisis de componentes principales

En el siguiente apartado se aborda el Análisis de Componentes Principales (PCA), una técnica empleada para la reducción de dimensionalidad en análisis de datos y aprendizaje automático.

PCA permite transformar un conjunto de variables correlacionadas en un nuevo conjunto de variables no correlacionadas, denominadas componentes principales. Estos componentes capturan la máxima variabilidad presente en los datos, facilitando así la interpretación y visualización de las relaciones subyacentes.

Es el último paso antes de comenzar con el desarrollo del algoritmo, y con él se da por concluidas las tareas relativas a la Fase I del proyecto.

Figura 14: Flujo de trabajo Fase II [PCA]

La mayoría de las plataformas consultadas por los *riders* para obtener predicciones meteorológicas ofrecen las siguientes características:

- Velocidad del viento
- Velocidad de la racha
- Dirección viento
- Altura oleaje
- Periodo oleaje
- Dirección del oleaje

Por lo tanto, se ha decidido prescindir de las características meteorológicas que no estén incluidas en la lista mencionada, ya que inicialmente no será posible proporcionarlas como entrada al algoritmo objetivo de este estudio.

Para realizar el proceso de PCA, el código usará las librerías `sklearn.decomposition` de forma iterativa, hasta obtener el menor número de variables que expliquen al menos el 95% de la varianza acumulada.

A continuación, se presentan gráficas que ilustran la evolución del conjunto de datos y sus varianzas en cada iteración del algoritmo de PCA.

La figuras 16, 20, 24 y 28 muestran la varianza explicada acumulada por cada componente principal e iteración. Estos valores indican cuánta varianza de los

datos originales es explicada por cada componente principal en orden de importancia.

Para decidir cuántos componentes principales conservar, generalmente se considera un valor acumulado que cubre un porcentaje alto de la varianza total. Un criterio común es conservar componentes que acumulen al menos el 95% de la varianza explicada. Este será el criterio que se evaluará en cada iteración hasta que no se puedan reducir más componentes.

Las figuras 15, 19, 23 y 27 muestran un mapa de calor de las variables y su carga de importancia en cada uno de los componentes tratados en cada iteración.

Las figuras 17, 21, 25 y 29 ponen en un eje cartesiano las cargas de las variables en las dos componentes que expliquen más la variabilidad del conjunto de datos.

En las figuras 18, 22, 26 y 30 muestran algunos datos estadísticos capturados por iteración en este proceso.

El proceso finaliza tras 3 iteraciones, determinando que es posible reducir el conjunto de datos a solo 5 variables:

- `peso_rider`
- `edad_rider`
- `periodo_medio_oleaje`
- `velocidad_viento`
- `racha`

Iteración 0 (Todos los componentes):

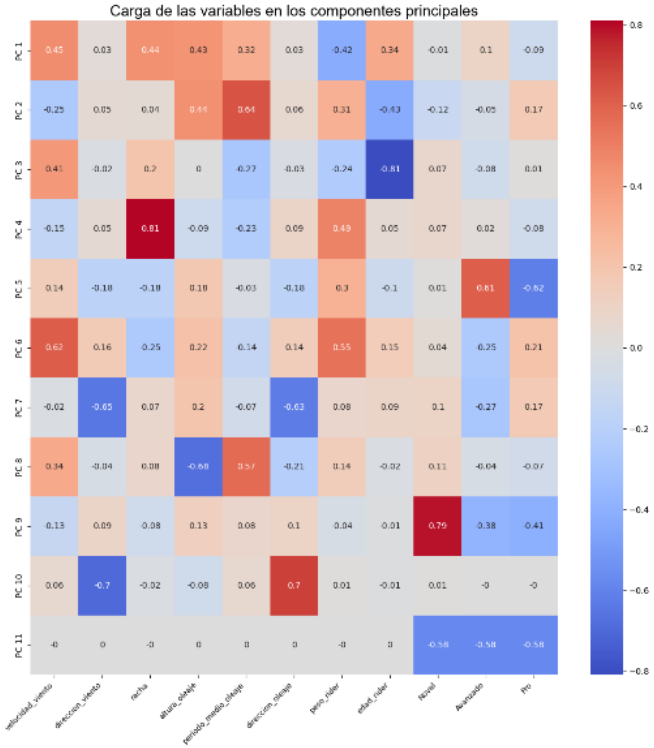


Figura 15: Mapa calor carga variables (PCA0)

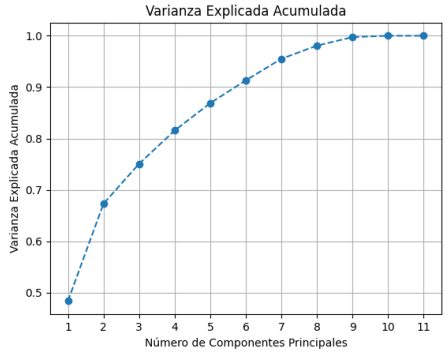


Figura 16: Varianza explicada acumulada (PCA0)

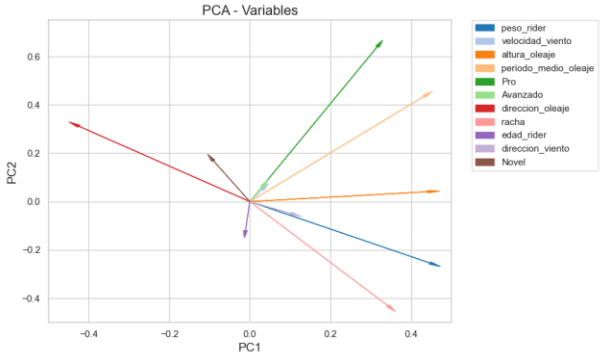


Figura 17: Relación variables con PC1 y PC2 (PCA0)

PCA0:

Las 11 variables más importantes: ['peso_rider', 'velocidad_viento', 'altura_oleaje', 'periodo_medio_oleaje', 'Pro', 'Avanzado', 'direccion_oleaje', 'racha', 'edad_rider', 'direccion_viento', 'Novel']

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11
Standard deviation	3.341917	1.308587	0.536631	0.446592	0.369128	0.305728	0.290010	0.176630	0.114528	0.018494	2.811067e-29
Proportion of Variance	0.483758	0.189424	0.077680	0.064646	0.053433	0.044256	0.041980	0.025568	0.016578	0.002677	4.069149e-30
Cumulative Proportion	0.483758	0.673182	0.750861	0.815508	0.868941	0.913196	0.955176	0.980745	0.997323	1.000000	1.000000e+00

Figura 18: Datos PCA0

Iteración 1:

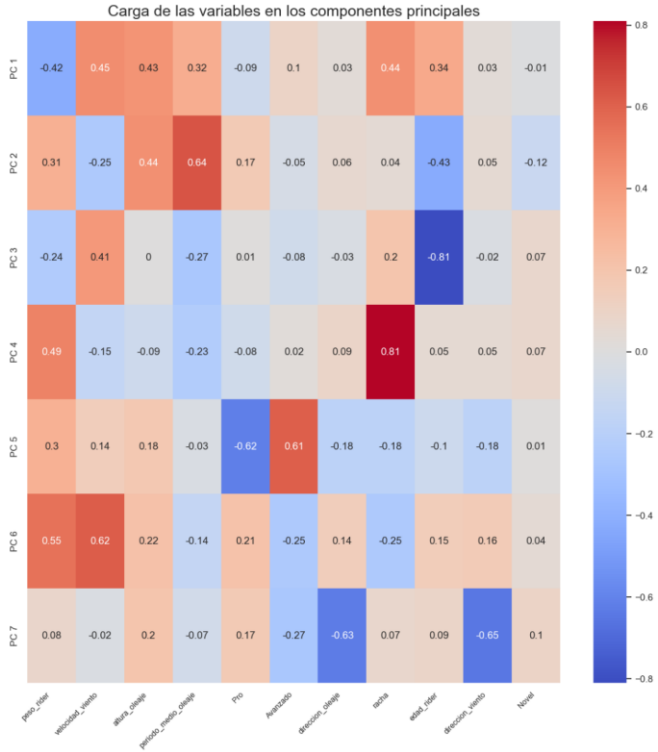


Figura 19: Mapa calor carga variables (PCA1)

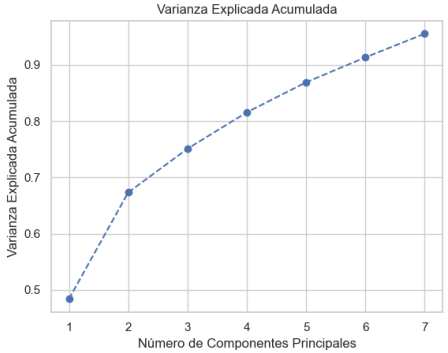


Figura 20: Varianza explicada acumulada (PCA1)

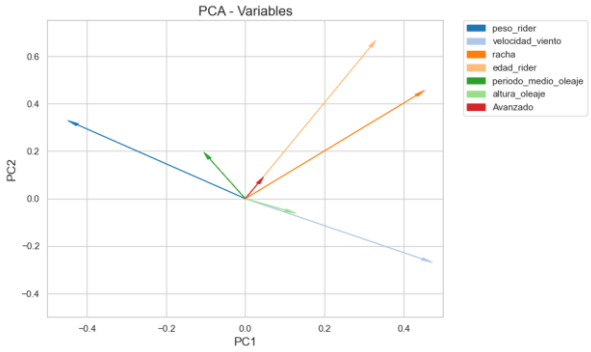


Figura 21: Relación variables con PC1 y PC2 (PCA1)

```
PCA1:
Las 7 variables más importantes: ['peso_rider', 'velocidad_viento', 'racha', 'edad_rider', 'periodo_medio_oleaje', 'altura_oleaje', 'Avanzado']
                            PC1       PC2       PC3       PC4       PC5       PC6       PC7
Standard deviation      3.309760  1.255251  0.533934  0.441464  0.314643  0.201772  0.176578
Proportion of Variance  0.530972  0.201375  0.085657  0.070822  0.050477  0.032369  0.028328
Cumulative Proportion   0.530972  0.732347  0.818004  0.888826  0.939303  0.971672  1.000000
```

Figura 22: Datos PCA1

Iteración 2:

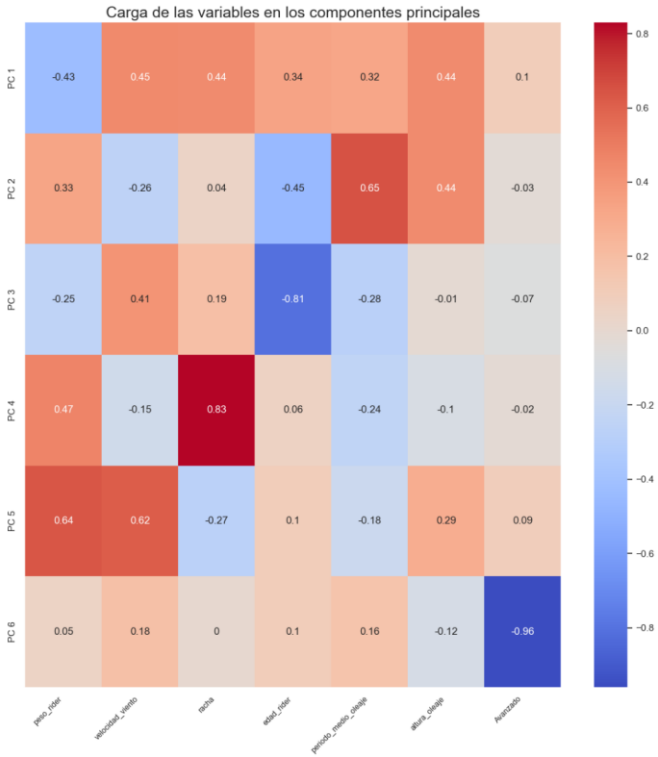


Figura 23: Mapa calor carga variables (PCA2)

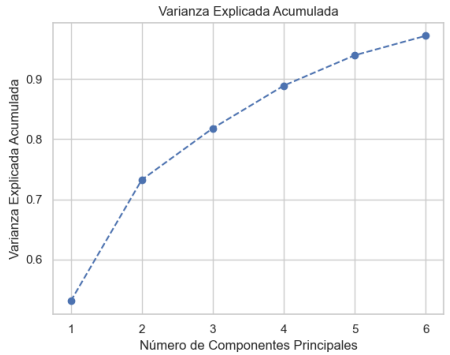


Figura 24: Varianza explicada acumulada (PCA2)

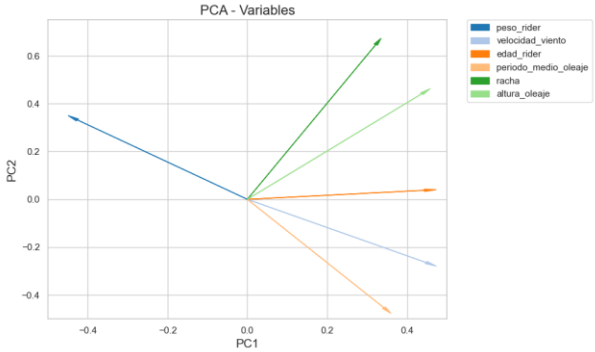


Figura 25: Relación variables con PC1 y PC2 (PCA2)

```
PCA2:
Las ó variables más importantes: ['peso_rider', 'velocidad_viento', 'edad_rider', 'periodo_medio_oleaje', 'racha', 'altura_oleaje']
                            PC1       PC2       PC3       PC4       PC5       PC6
Standard deviation      3.280548  1.254206  0.532455  0.441314  0.313656  0.178186
Proportion of Variance  0.546725  0.209022  0.088737  0.073548  0.052273  0.029696
Cumulative Proportion   0.546725  0.755746  0.844483  0.918031  0.970304  1.000000
```

Figura 26: Datos PCA2

Iteración 3:

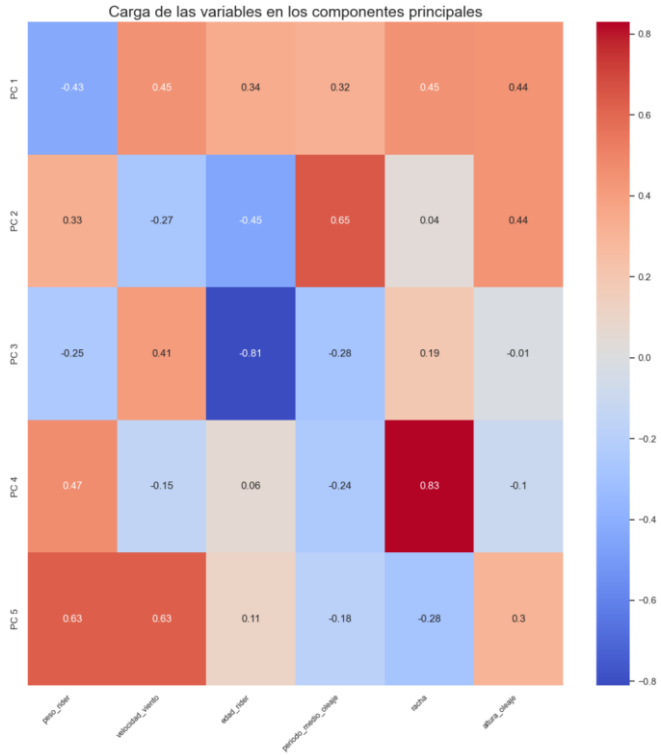


Figura 27: Mapa calor carga variables (PCA3)

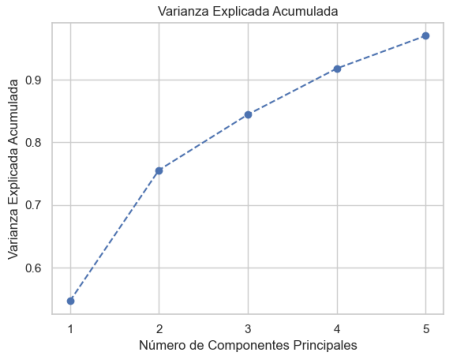


Figura 28: Varianza explicada acumulada (PCA3)

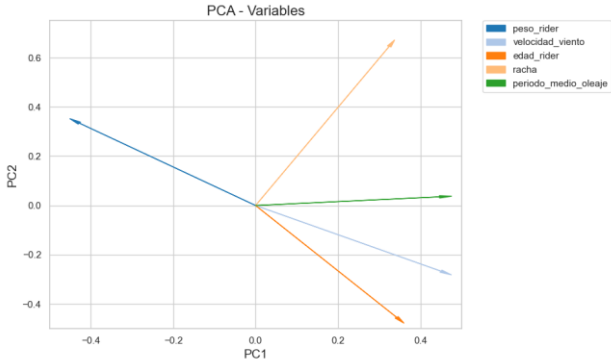


Figura 29: Relación variables con PC1 y PC2 (PCA3)

```
PCA3:
Las 5 variables más importantes: ['peso_rider', 'velocidad_viento', 'edad_rider', 'racha', 'periodo_medio_oleaje']
                           PC1       PC2       PC3       PC4       PC5
Standard deviation     2.756505  0.981079  0.532408  0.437487  0.292825
Proportion of Variance 0.551267  0.196204  0.106475  0.087492  0.058561
Cumulative Proportion  0.551267  0.747471  0.853946  0.941439  1.000000
```

Figura 30: Datos PCA3

Este proceso tiene como salida el fichero siguiente, que será el utilizado para el entrenamiento y validación del algoritmo final:

- `DATA/FIII_DATASET/f3_dataset.json`

9. Conclusiones Fase II

Los datos estudiados contienen las condiciones meteorológicas registradas en las estaciones de la AEMET y los registros modelados de los puntos SIMAR que afectan a la isla de Gran Canaria. Para obtener un conjunto de datos único, se han fusionado ambas fuentes y realizado algunas transformaciones y codificaciones.

Se ha establecido con éxito un proceso ETL robusto y bien estructurado para la recopilación, transformación y carga desde las dos fuentes principales: OpenData AEMET y Puertos.es. El uso de código Python ha facilitado la automatización de estas tareas, garantizando la eficiencia y precisión en el manejo de los datos.

Tras revisar los datos cargados, se observa que están bien estructurados y documentados. La limpieza de los datos ha sido adecuada, presentando una baja presencia de campos con valores nulos o vacíos. Además, muestran un alto potencial para derivar nuevos indicadores a partir de ellos.

Se ha introducido un formulario web para el etiquetado de datos, diseñado para la participación de los riders. Este formulario cuenta con un `frontend` en `React` y un `backend` con `Spring Boot`, desplegados en contenedores Docker. Se han realizado estrategias de promoción en redes sociales y centros locales para impulsar la participación. La recopilación de datos se considera muy exitosa en calidad y cantidad.

Se ha realizado una exploración detallada del conjunto de datos para comprender su estructura y calidad. Durante este proceso, se identificaron características que no aportaban significativamente a la información y que fueron eliminadas para simplificar el conjunto de datos y mejorar su calidad.

Además, se detectó un desbalanceo en las clases objetivo, lo que podría llevar a sesgos en los modelos de aprendizaje automático. Para abordar este problema, se realizó un balanceo adecuado de las clases, ajustando el tamaño medio de vela y tabla para las condiciones normales, lo que hizo que los datos fueran más representativos y ajustados a la realidad.

En cuanto a las correlaciones, se observó una fuerte relación positiva entre la velocidad del viento y la valoración, el peso del *rider* y el tamaño de la vela, así como entre el peso del *rider* y el tamaño de la tabla. Estos hallazgos sugieren que los *riders* más pesados suelen necesitar material más grande para las mismas condiciones que los más ligeros, y que la presencia de viento y olas mejora la experiencia de la sesión.

Por otro lado, se identificaron correlaciones negativas entre la velocidad del viento y el tamaño de la tabla y vela, y entre la valoración y el tamaño de la tabla. Estas relaciones indican que a mayor velocidad del viento, se prefieren tablas y velas más pequeñas, y que las tablas más pequeñas suelen conllevar a una mejor experiencia en la sesión.

En relación con los meses del año, se encontró que los meses de verano suelen ser los mejor valorados, con la excepción de julio. Meses como enero y febrero también destacaron en las valoraciones, posiblemente debido al mayor oleaje en comparación con los meses de verano.

Finalmente, el análisis de componentes principales reveló que es posible reducir las dimensiones del conjunto de datos a solo cinco, explicando más del 95% de la varianza inicial. Se destacó la importancia de variables como la edad en la elección del aparejo y la valoración de las condiciones, mientras que la altura del oleaje no resultó ser tan determinante como se podría haber pensado inicialmente.

10. Conjuntos *holdout set,* entrenamiento y prueba

El diseño adecuado de conjuntos de entrenamiento y prueba es fundamental en el desarrollo de modelos de *Machine Learning (ML)* como el que ocupa este trabajo, ya que influye directamente en la capacidad del modelo para generalizar patrones a datos no vistos previamente.

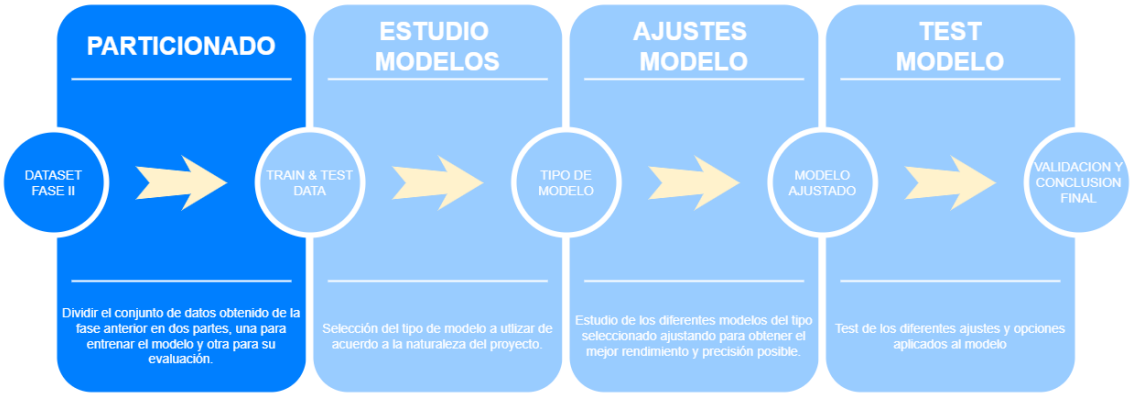


Figura 31: Flujo de trabajo Fase III [PARTICIONADO]

Para el presente proyecto se opta por dividir los datos obtenidos de los procesos anteriores en los conjuntos: *holdout set y* entrenamiento y validación cruzada en proporciones de 20% y 80% respectivamente del total. Esta estrategia optimiza tanto la validación del modelo como su capacidad para generalizar.

El *holdout set*, se utiliza para una evaluación final del modelo después del entrenamiento, garantizando que el modelo no se ha sobre ajustado a los datos de validación.

El conjunto de entrenamiento proporciona una cantidad adecuada de datos para que el modelo aprenda patrones y relaciones significativas subyacentes a los datos. Durante esta fase también se realiza la validación cruzada de la que se da más detalles en el apartado 10.2.

Esta división equilibrada asegura una evaluación robusta y objetiva del modelo, permitiendo una mayor confianza en la capacidad del modelo para generalizar a datos no vistos.

El proceso se realiza mediante la función `do_prep_train_test_data()` del fichero `TOOLS/ml_func.py`, que lee el fichero obtenido de la fase anterior (`DATA/FIII_DATASET/f3_dataset.json`) y genera dentro de `DATA/WC_TRAIN_TEST_DATA` los siguientes ficheros con las particiones mencionadas anteriormente que se usarán posteriormente:

- `train_df_val.csv` → Contiene los datos para entrenamiento y prueba estratificados por valoración.

- `train_df_sail.csv` → Contiene los datos para entrenamiento y prueba estratificados por tamaño vela.
- `train_df_wbs.csv` → Contiene los datos para entrenamiento y pruebas estratificados por tamaño de table.
- `test_df.csv` → Conjunto Holdout Set para la validación del algoritmo final.

El dominio de las variables objetivo se presenta como un continuo discretizado, lo que significa que son valores continuos limitados a un conjunto discreto de valores. Por tanto, una opción sería categorizar estas variables, pero se toma la decisión de tratarlas como continuas ya que conlleva diversas ventajas, entre las cuales se destacan:

- Mayor precisión en la representación de los datos.
- Conservación de la información original.
- Mejora en la capacidad de generalización.
- Flexibilidad en el análisis e interpretación.

10.1 Opciones del conjunto de datos

El algoritmo debe predecir 3 variables objetivo [`valoración, sail_size, wind_borard_size`], abordar el desafío de entrenar modelos para predecir múltiples variables objetivo en un conjunto de datos, surge la cuestión crucial sobre la obtención final de los datos para entrenamiento y prueba. Aquí, se exploran dos enfoques distintos junto con una estrategia incremental para abordar la complejidad inherente.

Opción 1: Conjunto de Datos Único: En esta opción, se elimina cada variable objetivo de los datos y se entrena un único modelo utilizando el conjunto resultante. Por ejemplo, si se está entrenando para predecir la valoración, la información sobre el tamaño de la vela y la tabla no se considera durante el entrenamiento.

Opción 2: Conjuntos de Datos Separados: Aquí, se eliminan solo las variables objetivo, creando así conjuntos de datos separados para entrenar modelos específicos para cada variable objetivo. Esto implica tener conjuntos de datos distintos para predecir el tamaño de la vela, el tamaño de la tabla y la valoración, respectivamente. Esta opción parece más completa ya que cada modelo considera todas las características relevantes.

No obstante, surge una complicación durante la ejecución: las entradas que se han utilizado para el entrenamiento no están disponibles, al menos para 2 de las 3 variables, ya que aún no han sido predichas. Esto podría reducir la precisión del modelo inicial.

Para abordar esta limitación, se propone una estrategia incremental. Se decide un orden para determinar las variables objetivo y se aumenta progresivamente el conjunto de datos de entrenamiento. Por ejemplo:

1. Conjunto para determinar valoración, contiene todas las variables excepto las tres variables objetivo.

2. Conjunto para determinar el tamaño de la vela: contiene todas las variables excepto el tamaño de tabla y la variable objetivo, o sea la valoración predicha se incluye como input.

3. Conjunto para determinar el tamaño de la tabla: contiene todas las variables, exceptuando la objetivo e incluyendo las predichas en los pasos 1 y 2.

Esta estrategia permite incorporar gradualmente la información predicha por los modelos anteriores en el proceso de entrenamiento, mitigando así las limitaciones de la opción 2.

En contrapartida, la estrategia descrita puede llevar a una acumulación significativa de errores en la predicción del "tamaño de vela" y el "tamaño de tabla". Cada error en las predicciones de etapas anteriores puede amplificar los errores en las etapas posteriores, lo que puede resultar en un rendimiento decreciente del modelo conforme se avanza en las predicciones secuenciales.

En la evaluación de los modelos se estudiarán ambas opciones descritas para determinar qué modelo y opción es el más adecuado.

10.2 Validación cruzada

Este enfoque implica dividir repetidamente el conjunto de datos en diferentes subconjuntos de entrenamiento y prueba, entrenando y evaluando el modelo en cada división y promediando los resultados.

Entre sus ventajas destacan:

- Mejora la generalización del modelo proporcionando una mejor estimación del rendimiento del modelo en datos no vistos.

- Reduce el posible sesgo de selección de datos ya que evalúa el modelo en múltiples particiones

- Utiliza de manera más eficiente el conjunto de datos disponible al aprovechar todas las observaciones tanto para entrenamiento como para prueba.

- Puede ayudar en la detección del sobreajuste, ya que realiza múltiples evaluaciones en diferentes conjuntos de datos.

Como contrapartida algunas de las desventajas al realizar una validación cruzada son:

- Puede ser computacionalmente costoso y requerir más tiempo de ejecución, especialmente con conjuntos de datos grandes.

- La implementación correcta de la validación cruzada puede ser más compleja y propensa a errores.

El conjunto de datos obtenido consta de 7616 registros con 8 características, con esta morfología está indicada una validación cruzada por los siguientes:

- El conjunto de datos es moderadamente grande, al permitir que cada punto de datos se utilice tanto para entrenamiento como para prueba en diferentes iteraciones, reduce la variabilidad en la estimación del rendimiento del modelo.

- Con 8 características, el modelo resultante puede ser relativamente complejo, la validación cruzada podría ayudar a evaluar como este generaliza con respecto a diferentes combinaciones de características y detectar si se está sobre ajustando los datos de entrenamiento.

- Se dispone de los recursos computacionales para realizar la generalización del modelo en tiempo razonable.

10.2.1 Número de pliegues de la validación cruzada

El número de pliegues en la validación cruzada se refiere a la cantidad de particiones en las que se divide el conjunto de datos durante el proceso de validación cruzada. Como se ha comentado anteriormente, cada partición se utiliza alternativamente como conjunto de entrenamiento y de prueba para evaluar el rendimiento del modelo.

Utilizar un mayor número de pliegues proporciona una estimación más estable del rendimiento del modelo, ya que el modelo se entrena y evalúa en múltiples subconjuntos de datos. Un número de pliegues muy alto tiene como contrapartida un mayor costo computacional, una menor velocidad de entrenamiento, una mayor variabilidad de resultados en conjuntos pequeños que difieren mucho y un posible sobreajuste del modelo.

Teniendo en mente el conjunto de datos que ocupa este proyecto y priorizando la estabilidad del rendimiento de modelo, se considera un número moderado de pliegues (5-10). Lo que debería proporcionar un buen equilibrio entre estabilidad y tiempo de computación. El número de pliegues se encuentra definido dentro del fichero `constants.py`.

```
# ML CROSS VALIDATION FOLDS
FOLDS = 5
```

En el siguiente apartado se realizarán varias simulaciones variando el número de pliegues comparando diferentes métricas de rendimiento y precisión para

determinar el número de pliegues que se adecúa mejor a los modelos estudiados.

11. Selección del modelo

La selección del modelo constituye una fase fundamental en proyectos de aprendizaje automático. Durante este proceso, se evalúan y comparan diversos modelos con el fin de determinar cuál se adapta mejor a los datos y objetivos específicos del proyecto, apoyándose en la validación cruzada.

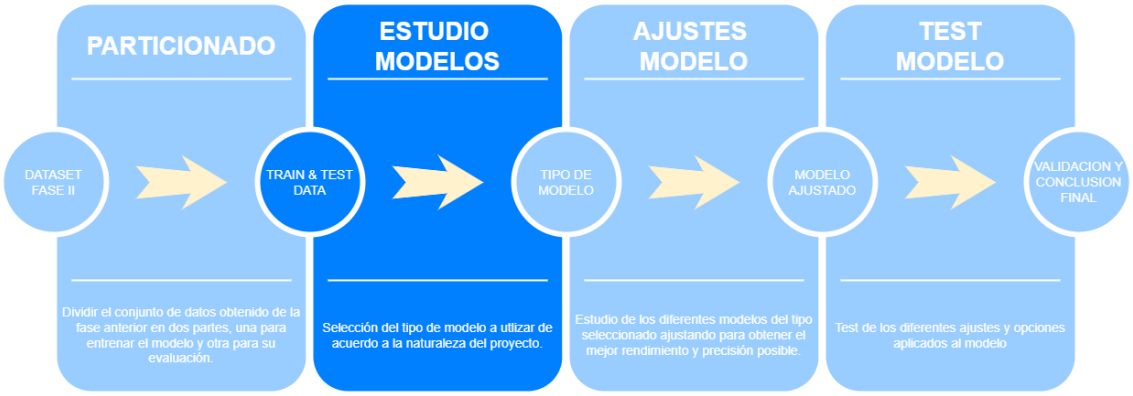


Figura 32: Flujo de trabajo Fase III [ESTUDIO MODELOS]

Para un proyecto como WindCaddy, donde todas las variables son continuas y se busca predecir el equipo óptimo basado en datos meteorológicos y características del *rider*, se considera clasificar como un problema de regresión.

11.1 Modelos evaluados

A continuación, se listan diferentes opciones de modelos de aprendizaje automático que se consideran para el proyecto que ocupa esta memoria:

11.1.1 Regresión Lineal Regularizada

Modelos como la Regresión Lineal, Regresión Ridge, LASSO y ElasticNet son conocidos por su capacidad para controlar el sobreajuste y generalizar bien, especialmente cuando hay muchas características y se quiere evitar la multicolinealidad. [8] [9] [10]

Regresión Lineal:

Es un modelo simple pero efectivo que podría funcionar bien si las relaciones entre las variables predictoras y la variable objetivo son aproximadamente lineales.

Regresión Ridge y Lasso:

Estos son modelos de regresión lineal regularizados que pueden ser útiles para manejar la multicolinealidad entre las variables predictoras y mejorar la generalización del modelo.

Para ambos modelos el término de regularización se denomina α y controla la cantidad de la fuerza de regularización para evitar el sobreajuste. Para el modelo

Ridge, el término de regularización es proporcional al valor absoluto de los coeficientes de las características, mientras que para el modelo Lasso, el término de regularización es proporcional al cuadrado de los coeficientes de las características.

El parámetro α se define dentro del fichero `constants.py`.

```
# RIDGE MODEL CONFIG
RALPHA = 1.0
# LASSO MODEL CONFIG
LALPHA = 1.0
```

Regresión ElasticNet:

ElasticNet es un modelo de regresión lineal regularizado que combina las penalizaciones de L1 (norma Lasso) y L2 (norma Ridge) en un solo término de regularización. Esta combinación permite obtener los beneficios de ambos enfoques y mitigar sus limitaciones individuales.

En este modelo α es una combinación ponderada de las penalizaciones L1 (Lasso) y L2 (Ridge). Se debe tomar en cuenta el parámetro L1 ratio, que controla la proporción de penalización L1 con respecto a L2. Es un valor entre 0 y 1, donde 0 indica que se utiliza sólo la penalización L2 y 1 indica que sólo se utiliza la penalización L1.

Ambos parámetros se definen dentro del fichero `constants.py`.

```
# ELASTICNET MODEL CONFIG
EALPHA = 0.1
EL1RATIO = 0.5
```

Todos estos modelos se encuentran dentro del módulo `sklearn.linear_model` y se utilizan las clases `LinearRegression`, `Ridge`, `Lasso` y `ElasticNet` respectivamente.

11.1.2 Support Vector Regression

Las SVM con kernel, especialmente las SVR con kernel RBF, pueden generalizar bien en una variedad de problemas de regresión, incluso cuando hay relaciones no lineales en los datos y hay una cantidad moderada a grande de datos de entrenamiento. [11] [12] [13] [14] [15]

Kernel Lineal:

El kernel lineal en SVM se utiliza cuando los datos son linealmente separables en el espacio de características actual.

Su término de regularización `C` es un parámetro que controla el equilibrio entre maximizar el margen y minimizar la clasificación incorrecta en el conjunto de datos de entrenamiento.

La elección de `C` depende de la complejidad del conjunto de datos y el compromiso deseado entre sesgo y varianza del modelo. Un valor más pequeño de `C` puede ayudar a prevenir el sobreajuste, especialmente cuando hay mucho ruido en los datos o cuando los datos no son linealmente separables. Por otro lado, un valor más grande de `C` puede ser más adecuado cuando se desea un modelo más complejo que se ajuste mejor a los datos de entrenamiento, aunque esto podría aumentar el riesgo de sobreajuste.

Kernel Polinomial:

El kernel polinomial es una función de kernel que calcula el producto interno de los vectores de características en un espacio de características de mayor dimensión generado por polinomios de grado específico.

Este kernel, se parametriza con los parámetros grado y gamma (ɣ), el primero indica el grado del polinomio a utilizar, a mayor grado mayor complejidad, permitiendo una mayor flexibilidad en la frontera de la decisión.

Los valores más altos de ɣ hacen que el modelo sea más sensible a los puntos de datos individuales y pueden conducir a modelos más complejos, mientras que los valores más bajos de ɣ hacen que el modelo sea menos sensible a los puntos de datos individuales y pueden dar lugar a modelos más suaves.

Kernel RBF (Radial Basis Function):

El kernel radial es una función de kernel que mapea los datos a un espacio de características de dimensionalidad infinita utilizando una función de base radial.

Gamma (ɣ) es un parámetro del kernel radial que controla la influencia de un solo ejemplo de entrenamiento. Valores más altos de gamma hacen que el modelo sea más sensible a los puntos de datos individuales, lo que puede llevar a modelos más complejos y ajustados a los datos de entrenamiento.

Por otro lado, valores más bajos de ɣ hacen que el modelo sea menos sensible a los puntos de datos individuales, lo que puede conducir a modelos más suaves y generalizados.

Tanto para RBF como para Polinomial, gamma (ɣ) puede tomar el valor '`scale`' y '`auto`', además de un `float` positivo.

Si el valor es '`scale`' se escala en función de la varianza de las características y el número de estas de acuerdo a:

$$\gamma = \frac{1}{nCaracterísticas * \sigma^2}$$

Si es 'auto', simplifica el cálculo y es útil cuando no se necesita ajustar gamma para cada conjunto de datos ya que el valor de gamma (ɣ) responde a:

$$\gamma = \frac{1}{nCaracterísticas}$$

Todos los parámetros de configuración para las SVM, se definen en el fichero constants.py.

```
# SVR MODEL CONFIG
SVRC = 0.01
SVRPDEGREE = 2
GAMMA = 'scale'
```

El modelo utilizado se encuentra en el módulo sklearn.svm y se utiliza la clase SVR [16].

11.1.3 Árboles de Decisión y Bosques Aleatorios

Los árboles de decisión y los bosques aleatorios tienen la capacidad de capturar relaciones no lineales y manejar interacciones complejas entre características. Son robustos frente al ruido en los datos y pueden generalizar bien, especialmente en conjuntos de datos grandes. [17] [18] [19] [20]

El modelo utilizado está dentro del módulo sklearn.esemble y se usa la clase RandomForestRegressor. [21]

RandomForestRegressor es un algoritmo de aprendizaje automático basado en árboles de decisión, que combina múltiples árboles para mejorar la precisión predictiva y reducir el sobreajuste.

Algunos de los hiperparámetros más importantes que afectan el rendimiento y la complejidad del modelo se listan a continuación:

n_estimators:
Determina el número de árboles en el bosque. Un mayor número de árboles generalmente lleva a un mejor rendimiento, pero también aumenta el costo computacional.

max_depth:
Controla la profundidad máxima de cada árbol en el bosque. Limitar la profundidad puede ayudar a prevenir el sobreajuste, especialmente en conjuntos de datos pequeños o ruidosos. Sin embargo, una profundidad demasiado baja puede llevar a un subajuste del modelo.

min_samples_split y min_samples_leaf:
Controlan el número mínimo de muestras requeridas para dividir un nodo interno o para ser considerado una hoja, respectivamente. Ajustar estos parámetros puede ayudar a regular la complejidad del modelo y prevenir el sobreajuste.

`max_features`:
Determina el número máximo de características a considerar al buscar la mejor división en cada nodo. Limitar este número puede mejorar la generalización del modelo y reducir la varianza, especialmente en conjuntos de datos con muchas características.

`bootstrap`:
Este parámetro indica si se deben tomar muestras de arranque al construir árboles. La selección de muestras de arranque ayuda a introducir variabilidad en los árboles y puede mejorar el rendimiento del modelo.

11.1.4 Gradient Boosting Machines

Los algoritmos de aumento de gradiente, como *Gradient Boosting Regressor*, son conocidos por su capacidad para construir modelos predictivos precisos y generalizar bien, especialmente cuando se ajustan cuidadosamente los hiperparámetros y se evita el sobreajuste. [22] [23] [24] [25]

Aunque esta biblioteca podría implementar la validación cruzada y muchos parámetros de ajuste de los conjuntos de entrenamiento, se opta por simplicidad, utilizar la misma lógica de particionado de estos que se usa para los demás modelos.

Se recurre a la biblioteca de código abierto `XGBoost`, donde `XGBRegressor` es la clase que se emplea [26] [27].

`XGBRegressor` es un modelo de regresión basado en árboles de decisión, específicamente en el algoritmo de Gradient Boosting.

Algunos de los hiperparámetros clave de este modelo son:

`n_estimators`:
Ídem que su homónima para `Random Forest`.

`max_depth`:
Ídem que su homónima para `Random Forest`.

`learning_rate`:
Controla la contribución de cada árbol al modelo y puede usarse para evitar el sobreajuste. Un valor más bajo requiere más árboles en el ensamble para lograr un rendimiento similar, pero puede mejorar la generalización del modelo.

`subsample`:
Determina la proporción de observaciones que se muestrean aleatoriamente para entrenar cada árbol. Un valor menor puede ayudar a evitar el sobreajuste y acelerar el entrenamiento, pero también puede reducir la precisión del modelo.

`colsample_bytree`:
Es la proporción de características que se muestrean aleatoriamente para entrenar cada árbol. Al igual que `subsample`, un valor menor puede ayudar a evitar el sobreajuste, especialmente cuando hay muchas características, pero también puede reducir la precisión del modelo.

`reg_alpha` y `reg_lambda`:
Son parámetros de regularización que controlan la penalización sobre los pesos de los árboles para evitar el sobreajuste.

`gamma`:
Controla la reducción mínima de la función de pérdida requerida para realizar una partición adicional en un nodo del árbol. Un valor más alto de este hiperparámetro hace que el algoritmo sea más conservador, lo que conduce a una mayor regularización.

11.1.5 Redes Neuronales Artificiales

Las redes neuronales, especialmente las arquitecturas más simples y reguladas adecuadamente, pueden generalizar bien en problemas de regresión, especialmente cuando hay una gran cantidad de datos y relaciones complejas entre las características. [28] [29] [30]

En este proyecto, para su implementación, se utiliza la biblioteca `tensorflow` de código abierto y desarrollada por Google. De las distintas opciones que ofrece `tensorflow` para la construcción del modelo, `tf.keras.sequential` es la más indicada para la resolución de modelos de regresión simple como el que ocupa este proyecto. [31]

La red neuronal definida en el proyecto es un modelo de regresión que utiliza tres capas totalmente conectadas:

- En la primera capa, se reciben las características de entrada y se procesan a través de 64 unidades con una función de activación `ReLU`, lo que permite que la red aprenda representaciones no lineales de los datos.

- Las características que provienen de la capa anterior se pasan a través de una segunda capa oculta con 32 unidades y la misma función de activación `ReLU`, lo que permite una mayor complejidad en la representación aprendida.

- En la capa de salida, se obtiene una predicción numérica, ya que solo hay una unidad sin una función de activación no lineal, lo que hace que la salida sea lineal.

La función de activación no lineal `ReLU`, utilizada en las dos primeras capas, retorna 0 si el valor de entrada es negativo o el valor de entrada si este es positivo.

Detalle del fichero `ml_func.py` con la definición de las capas de la red neuronal utilizada:

```
    model = tf.keras.Sequential([
        tf.keras.layers.Dense(64, activation='relu'),
        tf.keras.layers.Dense(32, activation='relu'),
        tf.keras.layers.Dense(1)  # Linear activation for regression
    ])
```

Este modelo busca minimizar la diferencia entre las predicciones y los valores reales de los datos de entrenamiento, ajustando los pesos de las conexiones entre las neuronas durante el proceso de entrenamiento mediante el algoritmo de retro propagación.

Los parámetros de ajuste de este modelo se definen en el fichero `constants.py`,

```
# NN MODEL CONFIG
OPTIMIZER = 'adam'
LOSS = 'mean_squared_error'
EPOCHS = 50
BATCH_SIZE = 32
VERBOSE = 0
```

Para problemas de regresión como el que nos ocupa, donde el objetivo es minimizar la diferencia entre las predicciones del modelo y los valores reales, `adam` es una opción sólida debido a su capacidad para ajustar la tasa de aprendizaje de manera adaptativa para cada parámetro. [32]

La función de pérdida de error cuadrático medio (`mean_squared_error`) calcula el cuadrado de la diferencia entre las predicciones del modelo y los valores reales, tomando el promedio de estos valores. Minimizar esta pérdida implica que las predicciones del modelo se acerquen lo más posible a los valores reales, lo que es justo lo que se pretende en un problema de regresión.

Se ha estimado que un valor de 50 épocas (`EPOCHS`) proporciona suficiente tiempo para que el modelo aprenda patrones complejos en los datos de entrenamiento y ajuste sus pesos en consecuencia, sin excederse en el entrenamiento y sobreajustando los datos.

El número de muestras por lote (`BATCH_SIZE`) para calcular el gradiente y actualizar los pesos en cada época, debe proporcionar una buena relación entre estabilidad y eficiencia computacional, un valor de 32 es una opción común y se estima conveniente para la implementación en windcaddy.

La cantidad de información de salida se ajusta con el parámetro `VERBOSE`, parametrizándose a `0` para evitar que se muestren detalles del entrenamiento.

11.2 Evaluación de los modelos

Dado que el objetivo es proporcionar sugerencias personalizadas basadas en condiciones específicas, será esencial evaluar estos modelos utilizando métricas de rendimiento adecuadas para la regresión.

Las métricas consideradas para la evaluación de los modelos estudiados serán el error cuadrático medio (`MSE`), el error absoluto medio (`MAE`) y el coeficiente de determinación (R^2). Idealmente, se busca un modelo con un `MSE` bajo, un `MAE` bajo y un R^2 alto. [33]

Los resultados de la evaluación de los diferentes modelos usando las opciones descritas en el apartado 10.1 y las métricas mencionadas anteriormente se muestran en las tablas sucesivas.

11.2.1 Opción 1: Conjunto de Datos Único

Variable [`valoracion`]:

***** valoracion *****

Model	MSE	MAE	R2	TIME ELAPSED
LinearRegression	0.324579	0.443366	0.693195	0.011002
ElasticNet	0.328922	0.446334	0.689211	0.014992
Ridge	0.324579	0.443371	0.693196	0.011000
Lasso	0.498701	0.529902	0.529453	0.014007
SVR Kernel: Linear	0.350571	0.429620	0.669139	0.599678
SVR Kernel: Polinomic	0.347198	0.417360	0.672488	0.429718
SVR Kernel: RBF	0.363998	0.419679	0.656697	0.622034
Random Forest	0.210525	0.234604	0.800856	0.993134
XGBRegressor	0.232232	0.274022	0.780456	1.117563
Neural Network	0.229919	0.311352	0.782691	12.657440

Figura 33: Predicción [`valoracion`] métricas modelos

Las 3 posibles alternativas para predecir esta variable entre los modelos evaluados son: `Random Forest, XGBRegressor y Neural Network`, en este orden. `Random Forest` parece la más indicada a tenor de su valor R^2 y los errores `MSE` y `MAE` son los más bajos, lo que indica que es realmente efectiva.

Variable [`sail_size`]:

```
*****  Op1:  sail_size  *****
                Model      MSE      MAE       R2  TIME ELAPSED
     LinearRegression 0.092426 0.220942 0.818620      0.011999
           ElasticNet 0.095571 0.224070 0.812283      0.017137
                Ridge 0.092426 0.220940 0.818620      0.011000
                Lasso 0.146604 0.300736 0.711543      0.015000
   SVR Kernel: Linear 0.094189 0.218742 0.815201      0.819767
SVR Kernel: Polinomic 0.097160 0.227922 0.809266      0.465384
      SVR Kernel: RBF 0.076329 0.188500 0.850253      0.520288
        Random Forest 0.032874 0.059072 0.935889      0.809410
         XGBRegressor 0.037683 0.081327 0.926293      0.213368
       Neural Network 0.056045 0.161890 0.890045     13.441043
```

Figura 34: Predicción [`sail_size`] métricas modelos (Opc. 1)

Para esta variable, `XGBRegressor` y `Random Forest`, tienen valores muy ajustados entre sí, aunque el rendimiento temporal de `XGBRegressor` es muy superior, por lo que en principio este modelo parece el más óptimo.

Variable [`wind_board_size`]:

```
*****  Op1:  wind_board_size  *****
                Model       MSE      MAE       R2  TIME ELAPSED
     LinearRegression 21.119639 3.393789 0.839769      0.012003
           ElasticNet 21.155220 3.381396 0.839532      0.016998
                Ridge 21.119620 3.393660 0.839769      0.011997
                Lasso 22.294124 3.417004 0.831007      0.014026
   SVR Kernel: Linear 23.053541 3.230588 0.825173      0.513290
SVR Kernel: Polinomic 24.761714 3.396294 0.812342      0.443716
      SVR Kernel: RBF 26.355811 3.634992 0.800328      0.852486
        Random Forest 11.908146 1.840293 0.909782      1.227462
         XGBRegressor 13.083346 2.232848 0.900877      0.295776
       Neural Network 14.678796 2.800250 0.889318     14.425355
```

Figura 35:Predicción [`wind_board_size`] métricas modelos (Opc. 1)

Las posibles alternativas para predecir esta variable entre los modelos evaluados son: `Random Forest` y `XGBRegressor` (valores $R^2 > 0.9$). `Random Forest` parece la más indicada, especialmente porque tiene el `MSE` más bajo y el `R2` más alto, lo que indica una mejor capacidad de predicción.

11.2.2 Opción 2: Conjuntos de Datos Separados:

Variable [`sail_size`]:

```
*****  Op2:  sail_size  *****
                Model      MSE      MAE       R2  TIME ELAPSED
     LinearRegression 0.084622 0.213590 0.834055      0.020999
           ElasticNet 0.089777 0.217725 0.823737      0.014000
                Ridge 0.084622 0.213583 0.834056      0.012001
                Lasso 0.146604 0.300736 0.711543      0.021000
   SVR Kernel: Linear 0.086533 0.210837 0.830334      0.832741
SVR Kernel: Polinomic 0.093552 0.224826 0.816410      0.421749
      SVR Kernel: RBF 0.071767 0.184067 0.859269      0.557409
        Random Forest 0.019302 0.027050 0.962418      0.928539
         XGBRegressor 0.019009 0.032841 0.963103      0.232753
       Neural Network 0.039380 0.130931 0.922553     13.737381
```

Figura 36: Predicción [sail_size] métricas modelos (Opc. 2)

Los modelos basados en árboles, especialmente `Random Forest` y `XGBRegressor`, muestran un rendimiento excepcionalmente alto con un `MSE` muy bajo y un R^2 muy alto. Destaca la rapidez en el aprendizaje con `XGBRegressor`, aproximadamente un 75% más rápida que `Random Forest`.

La red neuronal también sigue siendo una opción sólida, aunque su `MSE` y R^2 son ligeramente más altos que los modelos de árboles.

Variable [`wind_board_size`]:

```
*****  Op2:  wind_board_size  *****
                Model       MSE      MAE       R2  TIME ELAPSED
     LinearRegression 17.093988 3.119550 0.870181      0.018999
           ElasticNet 17.333086 3.103791 0.868457      0.017840
                Ridge 17.093641 3.119241 0.870185      0.012121
                Lasso 20.167626 3.201297 0.847135      0.016005
   SVR Kernel: Linear 19.868359 3.032752 0.849341      0.610661
SVR Kernel: Polinomic 23.983365 3.368776 0.818265      0.510610
      SVR Kernel: RBF 25.901733 3.626171 0.803802      0.911613
        Random Forest  9.423943 1.534431 0.928319      1.379000
         XGBRegressor 10.122399 1.860026 0.923247      0.239060
       Neural Network 11.499625 2.557115 0.913341     14.218870
```

Figura 37: Predicción [`wind_board_size`] métricas modelos (Opc. 2)

Al igual que la variable anterior, `Random Forest` y `XGBRegressor`, se muestran como buenas opciones con un `MSE` muy bajo y un R^2 muy alto. Finalmente los valores de `MSE` y `MAE` más bajos hacen de `Random Forest` la mejor opción

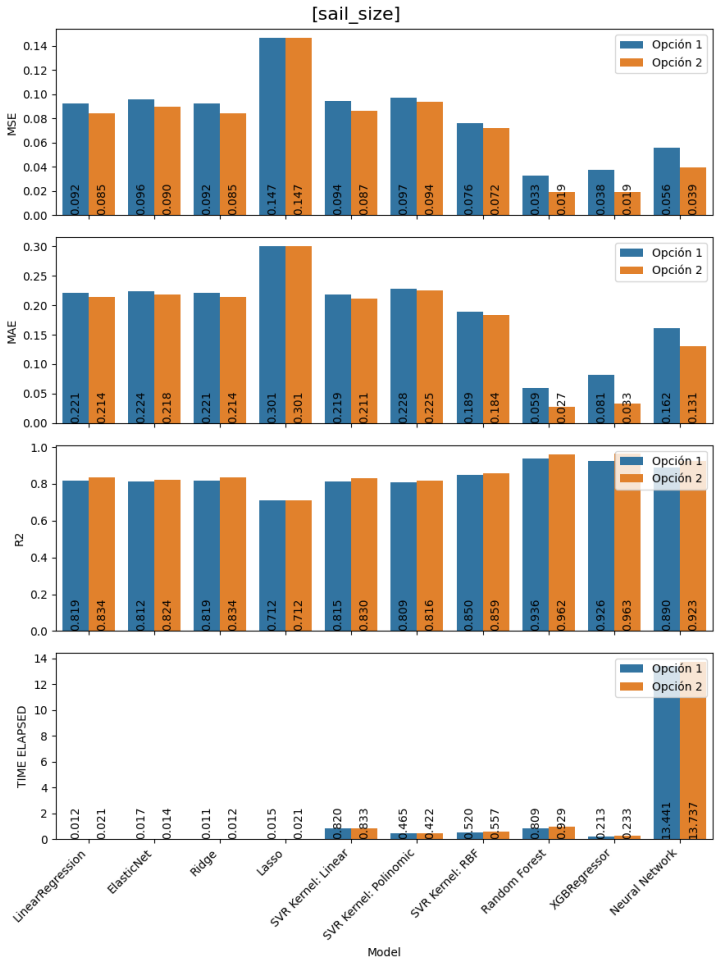


Figura 38: Comparativa modelos, opciones y métricas [`sail_size`]

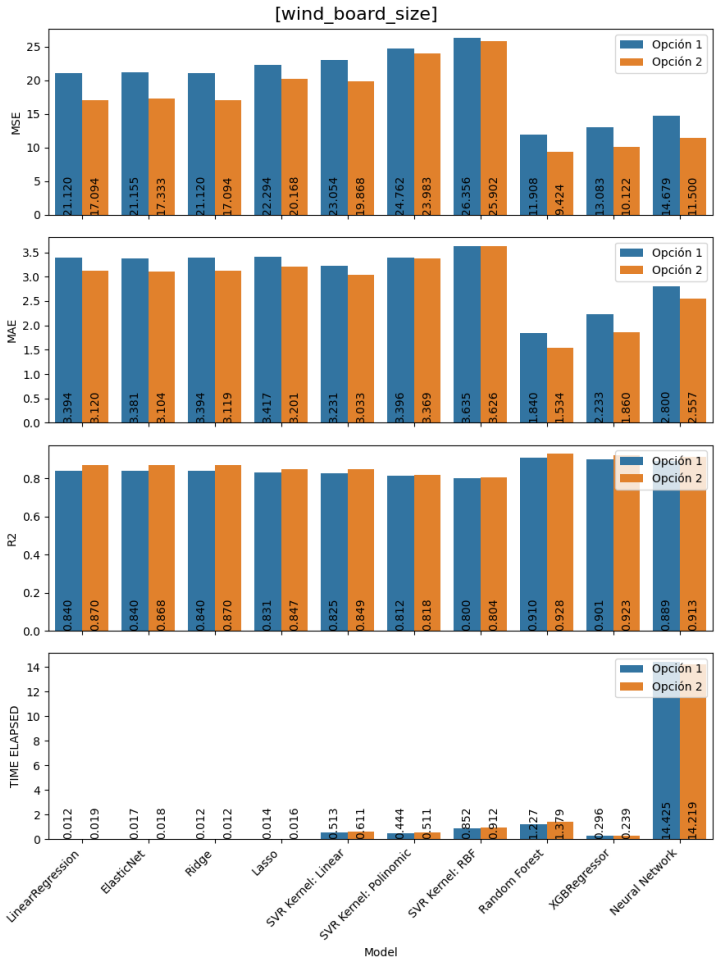


Figura 39: Comparativa modelos, opciones y métricas [`wind_board_size`]

11.3 Selección de los modelos a usar

En esta sección, se eligen los modelos finales que se utilizarán en función de los resultados evaluados previamente para cada una de las variables objetivo definidas.

Es importante tener en cuenta que el orden determinado para las distintas predicciones de las variables es: `valoracion`, `sail_size`, `wind_board_size`. Además, en cada predicción se utilizará como entrada el vector de las variables no objetivo más las variables predichas anteriormente.

11.3.1 Valoración

Esta variable será la primera en ser predicha por el algoritmo. Por lo tanto, la selección del modelo se fundamenta en lo discutido en el apartado 12.1.1 respecto a la variable de `valoracion`, donde se concluye que `Random Forest` es el modelo más apropiado

11.3.2 Tamaño de la vela

Como segunda variable a predecir, este modelo tomará como entrada el conjunto de variables no objetivo junto con la variable de `valoracion` predicha en el paso anterior. Para ello, se opta por la opción 2 descrita en el apartado 10.1 de esta memoria.

La selección del modelo para predecir esta variable se fundamenta en los resultados obtenidos en el apartado 12.1.2 referentes a la variable `sail_size`. En dicho análisis, se observó que los modelos basados en árboles proporcionaban las mejores métricas, aunque muy similares entre sí. Por consiguiente, la rapidez de `XGBRegressor` justifica la elección de este modelo para predecir la variable del tamaño de la vela.

11.3.3 Tamaño de la tabla

Como tercera y última variable a predecir, este modelo utilizará como entrada el conjunto de variables no objetivo, junto con las variables de `valoracion` y `sail_size` predichas anteriormente. Para ello, se opta por la opción 2 descrita en la apartado 10.1 de este documento.

Al igual que en el caso anterior, la elección del modelo para predecir esta variable se basa en los resultados obtenidos en el apartado 12.1.2 para la variable en cuestión. Se encontró que, nuevamente, los modelos basados en árboles ofrecían las mejores métricas, aunque muy próximas entre sí. Por tanto, la eficiencia de `XGBRegressor` justifica la selección de este modelo para predecir la variable del tamaño de la tabla.

12. Ajustes del modelo

En este apartado, se detallan los procesos llevados a cabo para optimizar y ajustar los hiperparámetros de los modelos seleccionados: `Random Forest` y `XGBRegressor`.

El objetivo principal de esta sección es maximizar el rendimiento predictivo de los modelos, garantizando al mismo tiempo su generalización y capacidad de adaptación a datos nuevos. Se exploran diversos enfoques de ajuste de hiperparámetros para encontrar la configuración óptima que mejore la precisión y la capacidad de generalización de cada modelo. [33]

Este proceso se realiza mediante las técnicas de búsqueda aleatoria y búsqueda en cuadrícula apoyada por una validación cruzada adecuada para evaluar el rendimiento de los modelos en conjuntos de datos no observados. Se usará el error cuadrático medio como métrica para evaluar los ajustes aplicados.

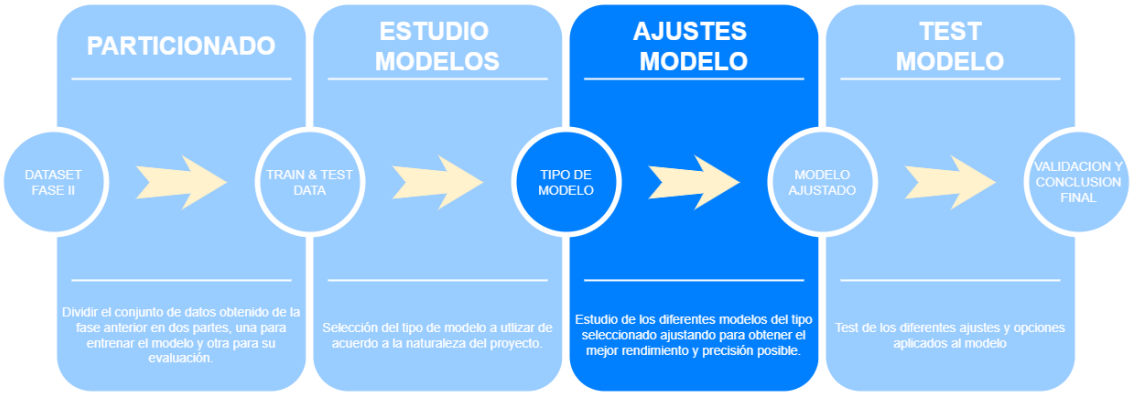


Figura 40: Flujo de trabajo Fase III [AJUSTES DEL MODELO]

12.1 Ajuste Bosque aleatorio

Se pretende encontrar la combinación óptima de hiperparámetros (véase apartado 11.1.3) que maximice el rendimiento predictivo del modelo, garantizando al mismo tiempo su capacidad de generalización.

Las cuadrículas que definen los espacios de búsqueda de hiperparámetros para este modelo, tanto para una la búsqueda aleatoria (`random`) como en cuadrícula (`grid`) se encuentra dentro de la función `do_tunning()` del fichero `ML/doML.py`.

```
    # Espacio de búsqueda Random Forest (Grid)
    param_grid_rf = {
        'n_estimators': [50, 100, 150],
        'max_depth': [None, 10, 20],
        'min_samples_split': [2, 5, 10],
        'min_samples_leaf': [1, 2, 4],
```

```
        'max_features': ['auto', 'sqrt']
    }
```

```
    # Espacio de búsqueda Random Forest (Random)
    param_random_rf = {
        'n_estimators': randint(50, 200),
        'max_depth': randint(2, 20),
        'min_samples_split': randint(2, 20),
        'min_samples_leaf': randint(1, 10),
        'max_features': ['auto', 'sqrt']
    }
```

12.1.1 Resultados de los ajustes

Se presentan los resultados para la búsqueda aleatoria y en cuadrícula:

```
**********************************************************************************************************

VARIABLE OBJETIVO: valoracion | MODELO: randomforest | BUSQUEDA TIPO: random

Mejores hiperparámetros:
max_depth: 12
max_features: sqrt
min_samples_leaf: 3
min_samples_split: 6
n_estimators: 160

Informe de métricas:
MSE: 0.107093
MAE: 0.175647
R2: 0.899085

Búsqueda finalizada en 00:00:03

**********************************************************************************************************
**********************************************************************************************************

VARIABLE OBJETIVO: valoracion | MODELO: randomforest | BUSQUEDA TIPO: grid

Mejores hiperparámetros:
max_depth: 20
max_features: sqrt
min_samples_leaf: 1
min_samples_split: 10
n_estimators: 150

Informe de métricas:
MSE: 0.098055
MAE: 0.168795
R2: 0.907602

Búsqueda finalizada en 00:00:07

**********************************************************************************************************
```

Figura 41: Resultados del ajuste para la variable `valoracion` y modelo `Random Forest`.

```
**********************************************************************************************************

VARIABLE OBJETIVO: wind_board_size | MODELO: randomforest | BUSQUEDA TIPO: random

Mejores hiperparámetros:
max_depth: 18
max_features: sqrt
min_samples_leaf: 5
min_samples_split: 17
n_estimators: 125

Informe de métricas:
MSE: 7.119492
MAE: 1.759744
R2: 0.946272

Búsqueda finalizada en 00:00:01

**********************************************************************************************************
**********************************************************************************************************

VARIABLE OBJETIVO: wind_board_size | MODELO: randomforest | BUSQUEDA TIPO: grid

Mejores hiperparámetros:
max_depth: 20
max_features: sqrt
min_samples_leaf: 1
min_samples_split: 5
n_estimators: 150

Informe de métricas:
MSE: 2.724797
MAE: 0.981857
R2: 0.979437

Búsqueda finalizada en 00:00:09

**********************************************************************************************************
```

Figura 42: Resultados del ajuste para la variable `wind_board_size` y modelo `Random Forest`.

12.2 Ajuste XGBRegressor

De manera similar al proceso anterior, el objetivo es encontrar la combinación óptima de hiperparámetros para el modelo en cuestión (véase apartado 11.1.4). Esto implica maximizar su rendimiento predictivo mientras se asegura su capacidad de generalización.

Las cuadrículas que definen los espacios de búsqueda de hiperparámetros para este modelo con búsqueda aleatorias, como en cuadrícula (`grid`) se encuentra dentro de la misma función `do_tunning()` del fichero `ML/doML.py`.

```python
    # Espacio de búsqueda XGBRegressor (Grid)
    param_grid_xgb = {
        'n_estimators': [100, 200, 300],
        'max_depth': [3, 5, 7],
        'learning_rate': [0.01, 0.1, 0.3],
```

```
        'min_child_weight': [1, 3, 5],
        'subsample': [0.6, 0.8, 1.0],
        'colsample_bytree': [0.6, 0.8, 1.0],
        'gamma': [0, 0.1, 0.2],
        'reg_alpha': [0, 0.1, 0.5],
        'reg_lambda': [0, 0.1, 0.5]
    }
```

12.2.1 Resultados de los ajustes

Se presentan los resultados para la búsqueda aleatoria y en cuadrícula:

```
**********************************************************************************************************

VARIABLE OBJETIVO: sail_size | MODELO: xgbregressor | BUSQUEDA TIPO: random

Mejores hiperparámetros:
colsample_bytree: 0.849587258196383
gamma: 0.0621760931849261
learning_rate: 0.03998006759305568
max_depth: 5
min_child_weight: 1
n_estimators: 164
reg_alpha: 0.224305855522751 67
reg_lambda: 0.036333901768962285
subsample: 0.9905113591449466

Informe de métricas:
MSE: 0.007281
MAE: 0.027779
R2: 0.985682

Búsqueda finalizada en 00:00:01

**********************************************************************************************************
**********************************************************************************************************

VARIABLE OBJETIVO: sail_size | MODELO: xgbregressor | BUSQUEDA TIPO: grid

Mejores hiperparámetros:
colsample_bytree: 0.8
gamma: 0
learning_rate: 0.1
max_depth: 5
min_child_weight: 1
n_estimators: 300
reg_alpha: 0.1
reg_lambda: 0.5
subsample: 0.8

Informe de métricas:
MSE: 0.001139
MAE: 0.016005
R2: 0.99776

Búsqueda finalizada en 00:11:51

**********************************************************************************************************
```

Figura 43: Resultados del ajuste para la variable `sail_size` y modelo `XGBRegressor`.

12.3 Conjunto final de hiperparámetros.

Después de aplicar los ajustes, se ha notado una mejora generalizada en los modelos seleccionados utilizando los parámetros obtenidos durante este proceso. A continuación, se presenta una tabla que ilustra el porcentaje de mejora en las métricas de los modelos con los hiperparámetros ajustados en comparación con las métricas iniciales.

TARGET	RANDOM SEARCH			GRID SEARCH		
	MSE	MAE	R^2	MSE	MAE	R^2
valoracion (Random Forest)	49.13%	25.13%	12.27%	53.42%	28.05%	13.33%
sail_size (XGBRegressor)	61.70%	15.41%	2.34%	94.01%	51.27%	3.60%
wind_boardsize (Random Forest)	24.45%	-14.68%	1.93%	71.09%	36.01%	5.51%

`Random Forest` en la variable "`valoración`":
Se logró una mejora significativa en las métricas `MSE`, `MAE` y `R2` utilizando la búsqueda en cuadrícula en comparación con la aleatoria. Esto sugiere que la búsqueda exhaustiva de hiperparámetros en cuadrícula fue más efectiva para optimizar el rendimiento del modelo en esta variable.

`XGBRegressor` en la variable "`sail_size`":
Ambos métodos de búsqueda mostraron mejoras en las métricas, la búsqueda en cuadrícula obtuvo un mejor desempeño en todas las métricas.

`Random Forest` en la variable "`wind_board_size`":
La búsqueda en cuadrícula superó a la búsqueda aleatoria en todas las métricas, mostrando una mejora más consistente y significativa en `MSE`, `MAE` y `R2`. Destaca que la búsqueda aleatoria obtiene una mejora negativa en `MAE` con respecto al algoritmo original.

A tenor de los resultados obtenidos, se estima que los parámetros para los modelos predictivos aplicados a cada variable objetivo que mejor se ajustan son los mostrados a continuación:

Valoración	Tamaño de vela	Tamaño de tabla
max_depth: 20 max_features: sqrt min_samples_leaf: 1 min_samples_split: 10 n_estimators: 150	colsample_bytree: 0.8 gamma: 0 learning_rate: 0.1 max_depth: 5 min_child_weight: 1 n_estimators: 300 reg_alpha: 0.1 reg_lambda: 0.5 subsample: 0.8	max_depth: 20 max_features: sqrt min_samples_leaf: 1 min_samples_split: 5 n_estimators: 150

13. Desarrollo y test del modelo final

En este apartado, se construirá una función que integre los tres modelos entrenados previamente. Esta función estará diseñada para recibir los parámetros `peso_rider`, `velocidad_viento`, `edad_rider`, `racha`, `periodo_medio_oleaje` y generar predicciones para las tres variables objetivo `valoración`, `sail_size` y `wind_borad_size`.

Se realizará una primera validación visual, sobre la validez de estas predicciones utilizando una muestra representativa de los datos de entrada obtenidos del proceso de etiquetado. En apartados posteriores se realiza una validación más exhaustiva de la misma.

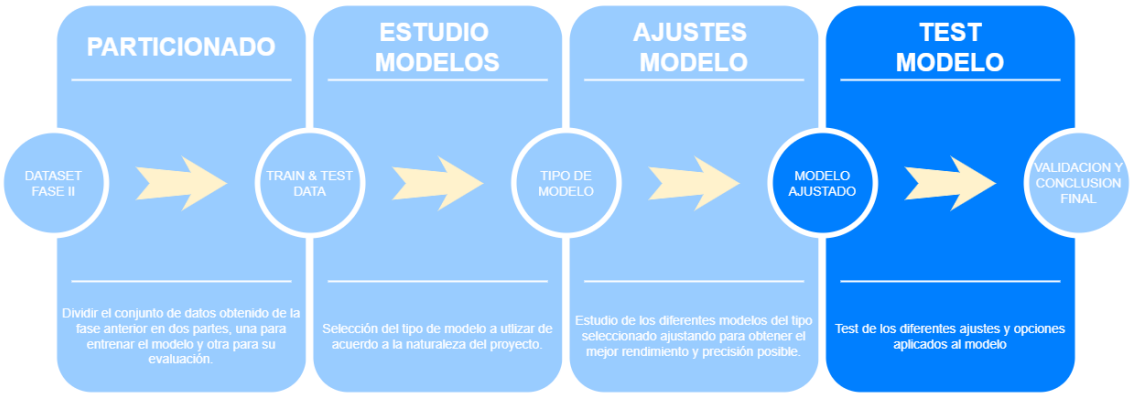


Figura 44:Flujo de trabajo Fase III [TEST MODELO]

13.1 Algoritmo final

La función objetivo de este proyecto (`windcaddyalg`), está diseñada para realizar evaluaciones de windsurf de forma automatizada. Toma como entrada el peso, la velocidad del viento, la edad, la racha y el periodo medio de oleaje.

Utiliza los modelos de *machine learning* previamente entrenados para predecir la valoración de la meteorología, así como el tamaño óptimo de vela y tabla para esas condiciones específicas.

En caso de que la valoración sea "Ni me tiro", indicando condiciones inadecuadas para windsurf, la función devuelve esta valoración junto con valores nulos para el tamaño de vela y tabla.

Puede consultarse el código Python de windcaddyalg en el anexo 21.2 de la presente memoria.

13.2 Extracción de la muestra de datos

Se extraen 10 registros aleatorios por valoración, de los datos etiquetados mediante consulta `SQL` a la base de datos etiquetados.

La consulta ejecutada puede consultarse en el anexo 21.3 del presente documento.

13.3 Validación visual

Se procede a validar visualmente los resultados de las predicciones de los datos extraídos del punto anterior. La ejecución de este proceso la realiza la función `do_validation()` del fichero `ML/doML.py`.

Por cada uno de los registros de entrada ofrece una salida como las que muestran a continuación:

```
Datos de entrada: [67 20 43 25  4]
Resultado: ('Ni me tiro', None, None)
Datos de entrada: [67 18 46 32  5]
Resultado: ('Me mojé, un rato', 4.1, 80.0)
Datos de entrada: [94 23 28 35  5]
Resultado: ('Buen baño', 6.2, 96.0)
Datos de entrada: [71 22 22 24  3]
Resultado: ('Me mojé, un rato', 5.5, 84.0)
Datos de entrada: [89 22 33 36  5]
Resultado: ('Buen baño', 6.1, 90.0)
```

Figura 45: Muestra de la validación visual de windcaddyalg

Tras esta primera valoración visual, se observa que los datos son coherentes con los resultados esperados, por lo que se da como una validación satisfactoria.

14. Conclusiones Fase III

Se han explorado diferentes enfoques para abordar la predicción de múltiples variables objetivo, optando por una estrategia incremental que aprovecha la información predicha por modelos anteriores. Además, se ha destacado la importancia de la validación cruzada para una evaluación robusta del rendimiento del modelo, proponiendo un número moderado de pliegues para equilibrar la estabilidad y la eficiencia computacional.

Tras evaluar diferentes opciones, se ha optado por `Random Forest` para predecir la valoración, `XGBRegressor` para el tamaño de la vela, y nuevamente `Random Forest` para el tamaño de la tabla. Estas decisiones se basan en las métricas de rendimiento obtenidas durante la evaluación de los modelos, asegurando así la calidad y eficacia de las predicciones para cada variable objetivo.

Tanto `Random Forest` como `XGBRegressor` fueron sometidos a ajustes exhaustivos utilizando técnicas de búsqueda aleatoria y en cuadrícula. Se lograron mejoras significativas en las métricas de rendimiento, como `MSE`, `MAE` y R^2, en comparación con los valores iniciales. Para ambos modelos, la búsqueda en cuadrícula fue más efectiva, mostrando mejoras significativas en todas las variables objetivo, decidiéndose utilizar los hiperparámetros propuestos en la búsqueda en cuadrícula para el algoritmo final.

Finalmente, se desarrolla la función `windcaddyalg`, integrando los modelos previamente entrenados para predecir la valoración meteorológica, tamaño de vela y tabla. Luego, se extraen 10 registros por valoración de la base de datos etiquetada con el objetivo de realizar una validación visual de las predicciones con los datos extraídos.

Se ha desarrollado un proceso robusto de predicción de múltiples variables objetivo, priorizando modelos como `Random Forest` y `XGBRegressor` tras ajustes exhaustivos. La implementación del algoritmo final `windcaddyalg` y la validación visual de las predicciones muestran que el enfoque propuesto ha sido óptimo.

15. Validación

Para la validación final del algoritmo desarrollado, se utilizarán datos reservados etiquetados que no han sido previamente vistos por el algoritmo. Este conjunto de datos, denominado *holdout set* en el apartado 10, corresponde al 20% del total de datos obtenidos.

Un *holdout set* es un subconjunto de datos que se separa del conjunto de datos original antes del entrenamiento del modelo y se reserva exclusivamente para evaluar su rendimiento. La finalidad de este conjunto es proporcionar una estimación imparcial de la capacidad del modelo para generalizar a nuevos datos. Evaluar el modelo utilizando el *holdout set* permite medir su desempeño en un contexto que simula escenarios reales, asegurando que el modelo no esté sobre ajustado a los datos de entrenamiento y que funcione correctamente en aplicaciones prácticas. [14] [34] [35]

15.1 Consideraciones en la validación

Durante todo el desarrollo del algoritmo, se ha considerado la predicción continua de las variables objetivo. Sin embargo, dado que el usuario final necesita recibir una valoración categórica, se transformará el resultado continuo en su equivalente categórico cuando sea necesario. Esta transformación se realizará utilizando la función `obtener_valoracion` del archivo `TOOLS/ml_func.py`.

Por lo anteriormente comentado, se hace necesaria la introducción de métricas adecuadas para la validación de variables categóricas. Métricas como precisión, sensibilidad, F1 score y exactitud están indicadas para evaluar modelos de clasificación con variables categóricas. La precisión y la sensibilidad proporcionan información sobre la calidad de las predicciones positivas, el F1 score equilibra ambos, y la exactitud ofrece una visión general del rendimiento. Estas métricas juntas permiten una evaluación completa y precisa del modelo. [29] [36]

15.1.1 Exactitud (Accuracy)

La exactitud mide la proporción de predicciones correctas (tanto verdaderos positivos como verdaderos negativos) sobre el total de predicciones realizadas.

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

La exactitud es útil cuando las clases están balanceadas, es decir, cuando el número de ejemplos en cada clase es aproximadamente el mismo. Proporciona una visión general del rendimiento del modelo.

15.1.2 Precisión (Precision)

La precisión mide la proporción de verdaderos positivos (TP) entre todos los ejemplos que el modelo ha etiquetado como positivos. En otras palabras, es la capacidad del modelo para no etiquetar como positivos los negativos.

$$Precisión = \frac{TP}{TP + FP}$$

Una alta precisión significa que la mayoría de las instancias etiquetadas como positivas son realmente positivas.

15.1.3 Sensibilidad (*Recall*)

El sensibilidd mide la proporción de verdaderos positivos entre todos los ejemplos que son realmente positivos. Indica la capacidad del modelo para encontrar todos los ejemplos positivos.

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

Una alta sensibilidad significa que el modelo captura la mayoría de los ejemplos positivos verdaderos.

15.1.4 F1 Score

El F1 score es la media armónica de la precisión y la sensibilidad. Proporciona un equilibrio entre ambos y es especialmente útil cuando las clases están desbalanceadas.

$$F1\ Score = 2\ \times \frac{Precisión \times Sensibilidad}{Presición + Sensibilidad}$$

El F1 score es alto cuando tanto la precisión como la sensibilidad son valores altos, y disminuye cuando uno de ellos es bajo.

15.1.5 Promedio Macro y Ponderado

Adicionalmente a las métricas evaluadas por clase, se calcula también los promedios Macro (*macro avg*) y Ponderado (*weighted avg*), estos pueden ser de utilidad en la evaluación de variables categóricas. [14] [37] [24]

El *Promedio Macro* calcula la métrica de interés (como precisión, sensibilidad y F1 score) para cada clase individualmente y luego toma el promedio aritmético

de esas métricas. Este método trata a todas las clases por igual, independientemente del número de ejemplos en cada clase.

$$Macro\ avg = \frac{1}{N}\sum_{i=1}^{N} Métrica_i$$

Donde N es el número de clases y $Métrica_i$ es la métrica calculada para la i-ésima clase.

Es útil cuando se quiere tener en cuenta el rendimiento del modelo en todas las clases de manera equitativa, independientemente del tamaño de las clases.

El promedio ponderado calcula la métrica de interés para cada clase y luego toma el promedio ponderado de esas métricas, ponderando por el número de ejemplos en cada clase. Esto significa que las clases con más ejemplos tendrán más influencia en el cálculo final.

$$Weighted\ avg = \frac{1}{T}\sum_{i=1}^{N} \omega_i \cdot Métrica_i$$

Donde T es el total de ejemplos, ω_i es el peso de la i-ésima clase, y $Métrica_i$ es la métrica calculada para la i-ésima clase.

Es útil cuando las clases están desbalanceadas y se quiere tener una evaluación global que refleje la distribución real de las clases, aunque puede ocultar el mal rendimiento en clases minoritarias si las mayoritarias dominan el *dataset*.

15.2 Validación versión 1

A continuación, se procede a mostrar los resultados del algoritmo desarrollado hasta el momento. Este algoritmo trata todas las variables objetivo como continúas discretizadas, aunque por las necesidades descritas anteriormente, la predicción continua de valoración será categorizada y evaluada con métricas continuas y categóricas.

```
###### windcaddy v1 Results ######
Doing validation
Valoracion
Continuous Evaluation
MSE: 0.30272108843537415
MAE: 0.26598639455782314
R²: 0.720278972709935
Categorical Evaluation
Accuracy: 0.7517006802721088
              precision    recall  f1-score   support

           1       0.18      0.67      0.29         3
           2       0.91      0.97      0.94       840
           3       0.74      0.75      0.74       329
           4       0.17      0.32      0.22       120
           5       0.00      0.00      0.00       178

    accuracy                           0.75      1470
   macro avg       0.40      0.54      0.44      1470
weighted avg       0.70      0.75      0.72      1470

sail_size
Continuous Evaluation
MSE: 0.034828533557219904
MAE: 0.04677641022189001
R²: 0.9300034858609859

wind_board_size
Continuous Evaluation
MSE: 8.242798353909466
MAE: 1.5157750342935528
R²: 0.9381285706908401
```

Figura 46: Resultados validacion windcaddyalg version 1

15.2.1 Conclusiones de la validación

Valoración

La evaluación continua muestra que el modelo tiene un buen desempeño en términos de `MSE` y `MAE`, aunque no es perfecto. El `R²` indica que

aproximadamente el 72% de la variabilidad en los datos de valoración es explicada por el modelo, lo cual es razonablemente bueno.

Como categórica (transformada), la precisión en la predicción de esta variable es de aproximadamente un 75%, lo que en principio es un valor aceptable.

Analizando las métricas categóricas expuestas se infiere que el modelo tiene serios problemas al identificar las clases 1 (Ni me tiro) y 4 (Bañazo) y es incapaz de predecir correctamente la clase 5 (Para Pro's).

El promedio ponderado muestra que, en general, el modelo tiene un buen desempeño, pero el promedio macro revela que hay clases (particularmente las clases 4 y 5) donde el modelo no se desempeña bien.

Tamaño de vela

La evaluación continua de esta variable muestra un rendimiento excelente. El bajo `MSE` y `MAE` sugieren errores mínimos en la predicción de esta variable. Con un R^2 de 0.93, el modelo explica la gran mayoría de la variabilidad de los datos.

El modelo se muestra altamente preciso y eficaz para predecir el tamaño de la vela lo que lo hace altamente confiable en este aspecto.

Tamaño de tabla

La evaluación continua de esta variable muestra un rendimiento muy bueno. Considerando el rango de los valores del volumen de las tablas (65-120 litros), el `MSE` y el `MAE` son relativamente bajos indicando que los errores de predicción son pequeños en comparación con el rango total de valores posibles. Con un R^2 de 0.93, el modelo explica la gran mayoría de la variabilidad de los datos.

El modelo se muestra es eficaz para predecir el tamaño de la tabla, lo que hace que sea confiable en este aspecto.

15.2.2 Conclusión

Los resultados anteriormente expuestos indican que el modelo desarrollado es efectivo para predecir tanto la valoración categórica como los tamaños de vela y tabla, aunque hay margen para mejorar la precisión en las clases minoritarias de la valoración categórica.

15.3 Desarrollo de versiones v2 y v3

Tras las conclusiones obtenidas de la validación, se ha decidido adoptar un enfoque diferente para la predicción de la variable valoración. Como se ha explicado anteriormente, esta variable se predice inicialmente de forma continua y luego se convierte a categórica. Para mejorar la precisión en la predicción, se desarrollarán y evaluarán las versiones 2 y 3 del algoritmo.

Estas versiones se entrenarán utilizando la valoración como una variable categórica y aplicarán modelos predictivos adecuados para esta tarea. La diferencia entre las versiones 2 y 3 radicará en el conjunto de datos de entrada utilizado para predecir las variables de tamaño de vela y tabla, según se detalla en el apartado 10.1 del presente documento.

Los conjuntos de datos utilizados para el entrenamiento, pruebas y validación son exactamente los mismos que los utilizados en el desarrollo y validación de la versión anterior.

En cuanto a los modelos utilizados para la predicción categórica de la valoración son los siguientes:

Regresión Lógística

Utiliza una función logística para transformar una combinación lineal de las variables independientes en un valor de probabilidad entre 0 y 1. Esto permite clasificar observaciones en categorías discretas de manera eficaz. Su interpretación y cálculo de probabilidades la hacen especialmente útil para problemas de clasificación binaria y multiclase. [38] [39]

Árbol de decisión clasificatorios

Divide los datos en subconjuntos basados en las características más informativas, creando reglas de decisión simples y comprensibles. Cada nodo del árbol representa una característica, cada rama una decisión, y cada hoja una categoría. Esto permite manejar tanto variables numéricas como categóricas y capturar relaciones no lineales entre las características y la variable objetivo. [40] [41]

Bosques aleatorios clasificatorios

Los Bosques Aleatorios Clasificatorios son adecuados para predecir variables categóricas debido a su capacidad para combinar múltiples árboles de decisión y reducir el sobreajuste. Al construir una variedad de árboles con diferentes muestras y características, estos modelos ofrecen una alta precisión predictiva y manejan eficazmente la complejidad de las relaciones entre características y la variable objetivo. Además, son robustos ante datos ruidosos y no requieren supuestos sobre la distribución de los datos, lo que los hace versátiles y aplicables a una amplia gama de problemas de clasificación. [42] [43]

Clasificador XGBoost

Este clasificador es una elección sólida para predecir variables categóricas debido a su capacidad para manejar conjuntos de datos grandes y complejos, su eficacia en la identificación de patrones sutiles en los datos y su capacidad para minimizar el sobreajuste. Además, su algoritmo de *boosting* secuencial mejora gradualmente la precisión del modelo, mientras que su flexibilidad permite la optimización de hiperparámetros para adaptarse mejor a las características específicas del conjunto de datos. [44] [45]

15.3.1 Evaluación de los modelos

En este apartado se evalúa el rendimiento de los modelos listados para para predicción de la valoración como variable categórica con las métricas descritas anteriormente.

A continuación, se muestran los datos de esta evaluación y la matriz de confusión para cada modelo:

```
Model: Logical Regression
Categorical Evaluation
Accuracy: 0.6828859060402684
              precision    recall  f1-score   support

           1       1.00      1.00      1.00         1
           2       0.82      0.91      0.86       347
           3       0.43      0.36      0.39       127
           4       0.33      0.28      0.31        53
           5       0.51      0.41      0.46        68

    accuracy                           0.68       596
   macro avg       0.62      0.59      0.60       596
weighted avg       0.66      0.68      0.67       596

Cross-Validation Scores:
 [0.71666667 0.68067227 0.66386555 0.68907563 0.69747899]
Mean CV Score: 0.6895518207282914
```

```
Model: Decision Tree Classifier
Categorical Evaluation
Accuracy: 0.8204697986577181
              precision    recall  f1-score   support

           1       1.00      1.00      1.00         1
           2       0.93      0.94      0.94       347
           3       0.75      0.74      0.75       127
           4       0.34      0.32      0.33        53
           5       0.71      0.74      0.72        68

    accuracy                           0.82       596
   macro avg       0.75      0.75      0.75       596
weighted avg       0.82      0.82      0.82       596

Cross-Validation Scores:
 [0.775      0.75630252 0.76470588 0.80672269 0.78151261]
Mean CV Score: 0.7768487394957984
```

```
Model: Random Forest
Categorical Evaluation
Accuracy: 0.8489932885906041
              precision    recall  f1-score   support

           1       0.00      0.00      0.00         1
           2       0.93      0.97      0.95       347
           3       0.78      0.69      0.73       127
           4       0.52      0.43      0.47        53
           5       0.77      0.88      0.82        68

    accuracy                           0.85       596
   macro avg       0.60      0.59      0.59       596
weighted avg       0.84      0.85      0.84       596

Cross-Validation Scores:
 [0.775      0.79831933 0.83193277 0.79831933 0.81512605]
Mean CV Score: 0.8037394957983194
```

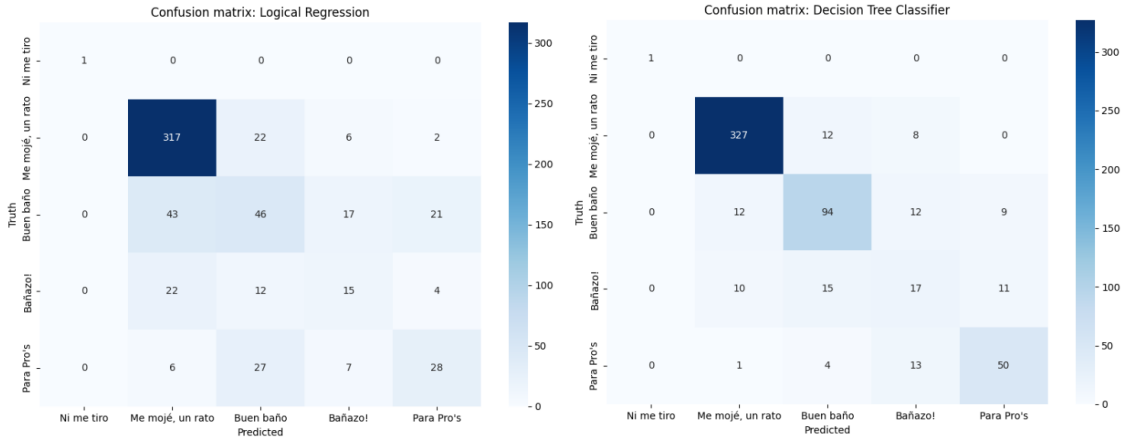
```
Model: XGB Classifier
Categorical Evaluation
Accuracy: 0.8305369127516778
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         1
           1       0.94      0.97      0.95       347
           2       0.71      0.71      0.71       127
           3       0.35      0.23      0.28        53
           4       0.73      0.81      0.77        68

    accuracy                           0.83       596
   macro avg       0.75      0.74      0.74       596
weighted avg       0.81      0.83      0.82       596

Cross-Validation Scores:
 [       nan 0.7394958  0.82352941 0.76470588 0.78991597]
Mean CV Score: nan
```

Figura 47: Métricas de valoración en los modelos categóricos evaluados



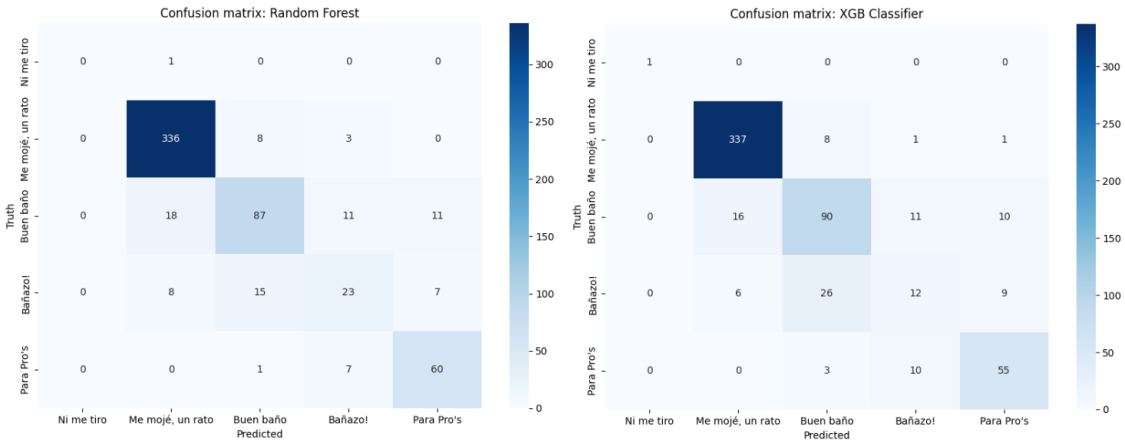


Figura 48: Matriz de confusión para valoración de los modelos evaluados

15.3.2 Selección del modelo

A tenor de los datos expuestos en el apartado anterior, se extraen las siguientes:

- La Regresión Logística tiene la menor precisión global (0.6829) y F1-Score ponderado (0.67). Aunque es un modelo simple y rápido de entrenar, su rendimiento no es suficiente.

- Árbol de Decisión mejora significativamente la precisión (0.8205) y el F1-Score (0.82), pero puede ser propenso al *overfitting* y no generaliza tan bien como los Bosques aleatorios.

- En comparación los Bosques aleatorios no solo tienen la mayor precisión (0.8490), sino también el mejor F1-Score ponderado (0.84) y el promedio más alto en la validación cruzada (0.8037), indicando un rendimiento sólido y consistente.

- `XGBoost` muestra buenos resultados (0.8305 de precisión y 0.82 de F1-Score), pero la falta de valores en la validación cruzada impide una evaluación completa de su estabilidad.

El modelo de Bosque aleatorio presenta altos valores de precisión y sensibilidad, especialmente para la clase mayoritaria (clase 2), pero también muestra un mejor rendimiento en comparación con los otros modelos en clases minoritarias (clase 5).

El F1-score es consistentemente alto en el modelo Bosque aleatorio, lo que indica un buen rendimiento general y balanceado.

El promedio de validación cruzada de Bosque aleatorio es el más alto (0.8037), lo que indica que el modelo es más robusto y menos propenso al *overfitting* comparado con los otros modelos.

Por lo que se determina que para las versiones 2 y 3 del algoritmo se usará Bosque Aleatorio para la predicción de valoración.

15.3.3 Desarrollo versión 2 y 3

En ambas versiones se usará el modelo de Bosque Aleatorio determinado en el apartado anterior para la predicción como categórica de la valoración.

En cuanto a la predicción de las variables continuas se ha decidido mantener los mismos modelos que en la versión 1, `XGBRegressor` para la predicción del tamaño de la vela y `RandomForestRegressor` para la predicción de la tabla.

15.4 Validación versión 2

```
###### windcaddy v2 Results ######
Doing validation
Valoracion
Categorical Evaluation
Accuracy: 0.9061224489795918
              precision    recall  f1-score   support

           1       1.00      0.67      0.80         3
           2       0.96      0.98      0.97       840
           3       0.85      0.82      0.83       329
           4       0.77      0.61      0.68       120
           5       0.84      0.93      0.88       178

    accuracy                           0.91      1470
   macro avg       0.88      0.80      0.83      1470
weighted avg       0.90      0.91      0.90      1470

sail_size
Continuous Evaluation
MSE: 0.022747101429460796
MAE: 0.04178595565660682
R²: 0.9542512883660869

wind_board_size
Continuous Evaluation
MSE: 8.069529652351738
MAE: 1.2992501704158146
R²: 0.9396252851193512
```

Figura 49: Resultados validación windcaddy version 2

15.4.1 Conclusiones

Valoración

La precisión de esta variable es alta, con aproximadamente un 90% de exactitud

Analizando las métricas expuestas se infiere que los mayores problemas que presenta el modelo y, aunque en todas las clases está por encima del azar, podrían presentarse en la predicción de las clases 4 (Bañazo!) y 1 (Ni me tiro).

Los promedios ponderado y macro muestran una mejora sustancial con respecto a la versión 1, en general todas las métricas en la versión 2 mejoran significativamente los de la versión anterior.

Tamaño de vela

La evaluación continua de esta variable muestra un rendimiento excelente. El bajo `MSE` y `MAE` sugieren errores mínimos en la predicción de esta variable. Con un R^2 de 0.95, el modelo explica la gran mayoría de la variabilidad de los datos.

El modelo mejora las métricas de la versión 1, mostrándose altamente preciso y eficaz para predecir el tamaño de la vela lo que lo hace altamente confiable en este aspecto.

Tamaño de tabla

La evaluación continua de esta variable muestra un rendimiento muy bueno. En comparación con la evaluación de la versión anterior y, aunque mejora su rendimiento, todas las métricas se mueven en unos valores muy similares por lo que no se considera que se mejore significativamente con respecto a la versión anterior en este aspecto.

Como la versión anterior, el modelo se muestra es eficaz para predecir el tamaño de la tabla, lo que hace que sea confiable en este aspecto.

15.5 Validación versión 3

```
###### windcaddy v3 Results ######
Doing validation
Valoracion
Categorical Evaluation
Accuracy: 0.9061224489795918
              precision    recall  f1-score   support

           1       1.00      0.67      0.80         3
           2       0.96      0.98      0.97       840
           3       0.85      0.82      0.83       329
           4       0.77      0.61      0.68       120
           5       0.84      0.93      0.88       178

    accuracy                           0.91      1470
   macro avg       0.88      0.80      0.83      1470
weighted avg       0.90      0.91      0.90      1470

sail_size
Continuous Evaluation
MSE: 0.03428084686165124
MAE: 0.03687799092288235
R²: 0.9310547507556792

wind_board_size
Continuous Evaluation
MSE: 9.004089979550102
MAE: 1.3476482617586911
R²: 0.9326330791638376
```

Figura 50: Resultados validacion windcaddy version 3

15.5.1 Conclusiones

Valoración

El modelo usado para la predicción de esta variable es el mismo que en la versión 2, por lo que las conclusiones del apartado anterior son aplicables a esta versión.

Tamaño de vela

En comparación con la versión 2, aunque se muestre con un `MAE` más bajo, los valores de `MSE` y R^2 indican que ofrece un menor rendimiento que su predecesor.

Tamaño de tabla

En esta variable no se muestra mejoría en ninguna de las métricas con respecto a la versión 2, lo que indica que en estas predicciones tiene un peor rendimiento que su predecesor.

16. Resultados

En este apartado se presentan los resultados obtenidos del análisis y evaluación de los modelos predictivos desarrollados para el proyecto WindCaddy. El objetivo de este es mostrar de forma resumida lo ya expuesto en apartados anteriores.

16.1 Descripción de los datos

El conjunto de datos utilizado en este estudio contiene datos meteorológicos y de condiciones del viento, así como información sobre el tamaño de la vela, el tamaño de la tabla y la valoración general de la sesión de deporte acuático. Los datos fueron preprocesados para eliminar valores faltantes y se dividieron en el conjunto para "entrenamiento y validación cruzada" y un "*holdout set*" que se usó para la validación final de las 3 versiones del algoritmo resultante.

16.2 Vector de entrada (inputs)

Debido a que son 3 las variables objetivo, se valoraron dos opciones para el entrenamiento y obtención del vector de entrada al algoritmo.

Opción 1	Opción 2
Mismo número de elementos en el vector de entrada para la predicción de todas las variables (los obtenidos del proceso de PCA).	Agregar las variables predichas con anterioridad al vector de entrada de las siguientes variables a predecir.

16.3 Modelos Evaluados

Se evaluaron cuatro modelos diferentes para cada variable objetivo:

En la versión 1 del algoritmo todas las variables fueron tratadas como continuas y los algoritmos usados para su predicción:

- Regresión Lineal (`LR`)
- ElasticNet (`EN`)
- Regresión Ridge (`RR`)
- Regresión Lasso (`RL`)
- Bosque Aleatorio regresivo (`RFR`)
- SVM (Kernel lineal) (`SVML`)
- SVM (Kernel polynomico) (`SVMP`)
- SVM (Kernel RBF) (`SVMR`)
- Regresión XGBoost (`XGBR`)
- Redes Neuronales (`NN`)

En las versiones 2 y 3 la variable valoración fue tratada como categórica desde el inicio y estos fueron los modelos usados para la predicción de esta variable

- Regresión Lógica (`RL`)
- Árbol de decisión de clasif. (`DT`)
- Bosque aleatorio de clasif. (`RFC`)
- XGBoost clasificador (XGBC)

Cada modelo fue entrenado y evaluado utilizando técnicas de validación cruzada para asegurar la robustez de los resultados.

16.4 Métricas de Evaluación

Las siguientes métricas se utilizaron para evaluar el rendimiento de cada modelo:

Métricas para variables continuas

- Error Cuadrático Medio (MSE)
- Error Absoluto Medio (MAE)
- Coeficiente Determinación (R^2)

Métricas en variables categóricas

- Exactitud (Accu.)
- Precisión (Prec.)
- Sensibilidad (Recall)
- F1 Score (F1 S)

16.5 Resultados

A continuación, se presenta una tabla resumen con los resultados de la evaluación de las versiones desarrolladas para cada variable objetivo.

Vers.	Variable	Mod.	MSE	MAE	R^2	Accu.	Preci. (M/W)	Recall (M/W)	F1 (M/W)
1	valoracion(*)	RFR	0.303	0.266	0.720	0.752	0.40 0.70	0.54 0.75	0.44 0.72
1	sail_size	XGBR	0.035	0.047	0.930	-	-	-	-
1	wind_board_size	RFR	8.243	1.516	0.938	-	-	-	-
2&3	valoracion	RFC	-	-	-	0.906	0.88 0.90	0.80 0.91	0.83 0.90
2	sail_size	XGBR	0.023	0.042	0.954	-	-	-	-
2	wind_board_size	RFR	8.070	1.299	0.940	-	-	-	-
3	sail_size	XGBR	0.034	0.037	0.931	-	-	-	-
3	wind_board_size	RFR	9.00	1.350	0.933	-	-	-	-

(*) Variable predicha como continua y convertida a categórica
(M/W) Valores de los promedios macro y ponderado de la métrica en las clases

16.6 Análisis de Resultados

Valoración (Categórica)

- v2 y v3: Ambos modelos tienen un rendimiento significativamente mejor que v1, con una exactitud de 90.61% comparada con el 75.17% de v1. Los promedios ponderados de precisión, sensibilidad y puntuación F1 también son superiores en v2 y v3.

Tamaño de la vela

- v2: Tiene el mejor rendimiento con el MSE más bajo (0.0227) y el R^2 más alto (0.9543).
- v3: Tiene un MAE más bajo que v2 y v1 (0.0369), pero su MSE y R^2 son peores que los de v2.

- v1: Tiene el peor rendimiento con el MSE más alto (0.0348) y el R^2 más bajo (0.9300).

Tamaño de la tabla

- v2: Tiene el mejor rendimiento con el MSE más bajo (8.0695), MAE más bajo (1.2993), y el R^2 más alto (0.9396).
- v1: Tiene un rendimiento ligeramente inferior a v2, con un MSE de 8.2428 y un R^2 de 0.9381.
- v3: Tiene el peor rendimiento con el MSE más alto (9.0041) y el R^2 más bajo (0.9326).

Rendimiento General

- Valoración: v2 y v3 son claramente superiores a v1.
- Tamaño de la vela: v2 tiene el mejor rendimiento general.
- Tamaño de la tabla de windsurf: v2 tiene el mejor rendimiento general.

La versión 2 del algoritmo ofrece el mejor rendimiento global, superando a v1 y v3 en todas las métricas importantes tanto para la valoración categórica como para la evaluación continua del tamaño de la vela y la tabla.

17. Conclusiones Fase IV y trabajos futuros

Este capítulo resume las conclusiones del proyecto WindCaddy, evalúa la consecución de los objetivos y la metodología utilizada, y analiza los impactos observados. Además, se proponen líneas de trabajo futuro para mejorar y ampliar el proyecto.

17.1 Descripción de las conclusiones del trabajo

En este proyecto, se desarrollaron y evaluaron varios modelos predictivos para ayudar a los deportistas acuáticos a seleccionar el equipo óptimo en función de las condiciones meteorológicas.

Los resultados obtenidos son alentadores y en gran medida coinciden con las expectativas, ya que se anticipaba que los modelos avanzados como Bosques aleatorios y `XGBoost` tendrían un mejor rendimiento debido a su capacidad para manejar datos no lineales y complejos.

Sorprende la capacidad del proceso de PCA para reducir la dimensionalidad del conjunto de datos inicial, eliminando variables que, a priori, parecían indispensables para el conjunto final de entrenamiento, como podría ser el *spot* o la altura de la ola.

17.2 Consecución de los objetivos planteados

Se han alcanzado todos los objetivos planteados inicialmente. Los modelos desarrollados lograron proporcionar predicciones precisas para la selección del equipo, optimizando así la experiencia y seguridad de los deportistas. La integración de técnicas de *Machine Learning* y análisis de datos avanzados permitió desarrollar un algoritmo confiable.
Aunque el resultado fue un algoritmo confiable, se estima que la premura en la obtención del conjunto de datos debido a los plazos de entrega ha afectado su variabilidad y calidad. El proceso de PCA dejó fuera variables que se consideraban importantes, como el nivel del *rider*, la altura de la ola e incluso el spot donde se practica. Esto sugiere que el proceso de recogida de datos no fue tan minucioso como se deseaba.

17.3 Metodología y seguimiento de la planificación

La planificación y la metodología previstas fueron adecuadas y se siguieron de manera rigurosa a lo largo del proyecto. No obstante, se realizaron algunos ajustes menores en las diferentes fases, los cuales están reflejados en los informes de seguimiento adjuntos a las entregas. Estos ajustes no afectaron ni a los hitos ni a los plazos de las entregas parciales.

17.4 Evaluación de impactos

Los impactos ético-sociales, de sostenibilidad y de diversidad mencionados en la sección 1.3 se han abordado adecuadamente. La plataforma WindCaddy

promueve la seguridad y la eficiencia en los deportes acuáticos, contribuyendo a una experiencia más segura y placentera para los usuarios. No se han identificado impactos negativos significativos durante el desarrollo del proyecto.

Tampoco se identificaron impactos no previstos durante el desarrollo del proyecto. La implementación de los modelos y la plataforma digital se llevó a cabo sin problemas significativos que afectaran negativamente el entorno o la sociedad.

17.5 Líneas de trabajo futuro

A pesar de los éxitos alcanzados, hay varias áreas de trabajo futuro que podrían explorar para mejorar y expandir WindCaddy:

1. **Mejoras en la precisión del modelo**: Aunque el algoritmo mostró un rendimiento excelente, explorar nuevos modelos podría mejorar el resultado final.

2. **Mejora de la obtención del conjunto de datos:** Mejorar el proceso de recogida de datos, asegurando una mayor variabilidad y calidad, para obtener un conjunto de datos más completo y representativo.

3. **Expansión de la localidad geográfica:** Los datos recopilados están limitados a los *spots* de la isla de Gran Canaria. Expandir geográficamente el conjunto de datos inicial para incluir otros *spots* de diferentes regiones y países, permitiría una mayor generalización y aplicabilidad del algoritmo a diversas condiciones y entornos acuáticos.

4. **Integración de datos en tiempo real**: La incorporación de datos meteorológicos en tiempo real podría mejorar la precisión y la utilidad de las predicciones, proporcionando recomendaciones más dinámicas y adaptadas al momento.

5. **Expansión a otros deportes acuáticos**: Ampliar la capacidad del algoritmo para incluir otros deportes acuáticos y sus respectivos equipos podría aumentar la utilidad de WindCaddy para una audiencia más amplia.

6. **Desarrollo de una aplicación móvil**: Crear una aplicación móvil permitiría a los usuarios acceder a las recomendaciones de equipo de manera más conveniente, directamente desde su dispositivo mientras están en el *spot* donde van a navegar.

7. **Análisis de impacto ambiental**: Integrar análisis de impacto ambiental para promover prácticas deportivas sostenibles y conscientes del entorno.

18. Leyendas y tablas explicativas

12.1 Evaluación del riesgo

La evaluación del nivel de riesgo se ha realizado sobre la base del siguiente producto entre su probabilidad y el impacto:

Impacto			
Alto	*Mediano*	Alto	Alto
Mediano	*Mediano*	*Mediano*	Alto
Bajo	*Bajo*	*Bajo*	*Mediano*
	Baja	*Mediana*	*Alta*

Probabilidad

12.2 Leyenda estados de tareas

Leyenda estados (Tareas)	
Por hacer	La tarea aún no ha sido iniciada, pero está planificada para realizarse en el futuro cercano.
En progreso	La tarea ha comenzado y actualmente está en curso de ejecución.
Pendiente de revisión	La tarea ha sido completada, pero está esperando ser revisada por un supervisor o responsable antes de ser marcada como finalizada.
En espera	La tarea está detenida temporalmente debido a algún motivo, como la espera de información adicional, recursos o decisiones externas.
En revisión	La tarea está siendo evaluada o revisada por alguien para su aprobación o corrección.
Finalizada	La tarea ha sido completada con éxito y ha alcanzado su objetivo.
Cancelado	La tarea ha sido interrumpida o anulada antes de su finalización, ya sea debido a cambios en los requisitos, falta de recursos o decisiones estratégicas.

12.3 Leyenda estados de hitos

Leyenda estados (Hitos)	
Planificado	El hito está programado para ser alcanzado en una fecha específica.
En progreso	Se está trabajando en la consecución del hito, pero aún no se ha completado.
Retrasado	El hito no se ha alcanzado en la fecha prevista y se ha pospuesto para una fecha posterior.
Alcanzado	El hito ha sido alcanzado con éxito según lo planeado.
Cancelado	El hito ha sido cancelado y ya no se espera que se alcance.

19. Glosario

B

backend
Backend
Es la capa de aplicación que se encarga de la lógica y manejar las solictudes del frontend. · 26

D

Dataset
Dataset Conjunto de datos · 11
Deadline
Deadline Fecha límite o plazo establecido para completar una tarea, proyecto o actividad específica. · 12

F

foils
Foil
Componente aerodinámico diseñado para crear sustentación en el agua y elevar el equipo sobre la superficie. · 1
freestyle
Freestyle
(Modalidad) Disciplina que se basa en realizar trucos y saltos en agua plana (sin olas). · 2
frontend
Frontend Capa de la aplicación que se ejecuta en el lado del cliente. · 26

K

Kick Off
Kick Off Acto de inicio del proyecto y sus actividades. · 11

R

rider
Rider
Deportista de deporte acuatico de deslizamiento. · 30

S

slalom
Slalom
(Modalidad) Disciplina basada en carreras en grupos durante un recorrido. · 2
spot
Spot
Lugar apto para la práctica del deporte a estudiar. · 4

W

waves
Waves
(Modalidad) Disciplina que se basa en realizar maniobras y saltos apoyado en olas. · 2

20. Referencias

[1] «Factorialhr.es,» 26 02 2024. [En línea]. Available: https://factorialhr.es/blog/coste-empresa-trabajador/.

[2] «Glassdoor.es,» 27 Febrero 2024. [En línea]. Available: https://www.glassdoor.es/Sueldos/index.htm.

[3] «Tusalario.es,» 27 Febrero 2024. [En línea]. Available: https://tusalario.es/carrera/funcion-y-sueldo/meteorologos.

[4] J. C. y. J. O. E. Guisado, «idus.us.es,» 2017. [En línea]. Available: https://idus.us.es/bitstream/handle/11441/89885/Geo_temas17%20%281%29.pdf?sequence=1&isAllowed=y#:~:text=Los%20puntos%20SIMAR%2C%20pertenecientes%20a,desde%201958%20hasta%20la%20actualidad.. [Último acceso: 28 02 2024].

[5] «StackExchange,» 13 03 2015. [En línea]. Available: https://stats.stackexchange.com/questions/141625/regression-and-correlation-of-wind-direction-circular-data. [Último acceso: 01 03 2024].

[6] «Wikipedia,» 03 12 2023. [En línea]. Available: https://en.wikipedia.org/wiki/Directional_statistics. [Último acceso: 01 03 2024].

[7] «Wikipedia,» 03 04 2023. [En línea]. Available: https://en.wikipedia.org/wiki/Pearson_correlation_coefficient#Circular. [Último acceso: 05 04 2024].

[8] J. Osborne, Regression & Linear Modeling: Best Practices and Modern Methods, SAGE Publications, Inc, 2017.

[9] P. Martin, Simple linear regression, SAGE Publications Ltd, 2021.

[10] D. B. &. L. K. Wright, Modern Regression Techniques Using R, SAGE Publications Ltd, 2009.

[11] P. Xiao, Artificial Intelligence Programming with Python, Wiley, 2022.

[12] «scikit-leanr.org,» 2024. [En línea]. Available: https://scikit-learn.org/stable/modules/svm.html. [Último acceso: 01 05 2024].

[13] K.-W. C. C.-J. H. X.-R. W. C.-J. L. Rong-En Fan, «LIBLINEAR: A Library for Large Linear Classification,» *Machine Learning Research,* nº 9, pp. 1871-1874, 2008.

[14] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[15] B. S. Alex J. Smola, «A tutorial on support vector regression,» *Statistics and Computing,* vol. 14, pp. 199-222, 2004.

[16] «scikit-learn.org,» 2024. [En línea]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html. [Último acceso: 2024 05 01].

[17] «berkeley.edu,» [En línea]. Available: https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm. [Último acceso: 02 05 2024].

[18] M. R. Segal, «Machine Learning Benchmarks and Random Forest Regression,» 2004. [En línea]. Available: https://escholarship.org/uc/item/35x3v9t4. [Último acceso: 02 05 2024].

[19] L. Breiman, «Random Forests,» *Machine Learning,* vol. 45, pp. 5-32, 2001.

[20] T. T. R. &. F. J. Hastie, The Elements of Statistical Learning Data Mining, Inference, and Prediction, Second Edition, Springer New York, 2009.

[21] «scikit-learn.org,» 2024. [En línea]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html. [Último acceso: 2024 05 01].

[22] «themachinelearners.com,» 2024. [En línea]. Available: https://www.themachinelearners.com/xgboost-python/. [Último acceso: 2024 05 01].

[23] C. Wade, Hands-On Gradient Boosting with XGBoost and scikit-learn, Packt Publishing, 2020.

[24] J. H. Friedman, «Greedy Function Approximation: A Gradient Boosting Machine,» *The Annals of Statistics,* vol. 29, nº 5, pp. 1189-1232, 2000.

[25] A. K. Alexey Natekin, «Gradient Boosting Machines: A Tutorial,» *Frontiers,* vol. 7, 4 12 2013.

[26] «xgbosst.readthedocs.io,» 2022. [En línea]. Available: [https://xgboost.readthedocs.io/en/stable/python/python_api.html#xgboost.XGBRegressor] . [Último acceso: 01 05 2024].

[27] «xgboost.readthedocs.io,» 2022. [En línea]. Available: https://xgboost.readthedocs.io/en/stable/. [Último acceso: 01 05 2024].

[28] Y. B. A. C. Ian Goodfellow, Deep Learning, Cambridge, Massachusetts : The MIT Press, 2016.

[29] C. M. Bishop, Neural Networks for Pattern Recognition, Clarendon Press · Oxford, 1995.

[30] Y. B. Y. &. H. G. LeCun, «Deep learning,» *Nature,* vol. 521, nº 7553, pp. 436-444, 2015.

[31] «tensorflow.org,» 10 01 2022. [En línea]. Available: https://www.tensorflow.org/guide/keras/sequential_model?hl=es-419. [Último acceso: 01 05 2024].

[32] «tensorflow.org,» 27 04 2024. [En línea]. Available: https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam. [Último acceso: 01 05 2024].

[33] A. Zheng, «Evaluating Machine Learning Models,» O'Reilly Media Inc., 2015.

[34] P. Domingos, «A Few Useful Things to Know about Machine Learning,» *Communications of the ACM,* vol. 55, nº 10, pp. 78-87.

[35] D. B. L. L. D. e. a. Krstajic, «Cross-validation pitfalls when selecting and assessing regression and classification models.,» *J Cheminform,* vol. 6, nº 10, 2014.

[36] E. F. y. M. A. H. Ian H. Witten, Data Mining: Practical Machine Learning Tools and Techniques, Elsevier, 2011.

[37] K. P. Shung, «Accuracy, Precision, Recall or F1?,» 15 03 2018. [En línea]. Available: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9. [Último acceso: 06 06 2024].

[38] C. Y. J. L. K. L. &. I. G. M. Peng, «An Introduction to Logistic Regression Analysis and Reporting.,» *The Journal of Educational Research,* vol. 96, nº 1, pp. 3-14, 09 2002.

[39] S. L. y. R. X. S. David W. Hosmer, Applied Logistic Regression, Wiley, 2013.

[40] L. y. M. O. Rokach, Data Mining and Knowledge Discovery Handbook, Springer, 2009.

[41] M. S. y. V. K. Pang-Ning Tan, Introduction to Data Minin, Pearson, 2005.

[42] L. Breiman, «Random Forests,» *Machine Learning,* vol. 45, nº 1, pp. 5-32, 2001.

[43] A. L. y. M. Wiener, Classification and Regression by Random Forest, Springer, 2001.

[44] T. y. G. C. Chen, «XGBoost: A Scalable Tree Boosting System,» de *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[45] H. Mittal, «Machine Learning with XGBoost Using R,» *Journal of Chemical Information and Modeling ,* vol. 53, nº 9, pp. 1689-1699, 2013.

21. Anexos

En esta sección de anexos se proporciona información complementaria y detallada relativa al proyecto. Los anexos incluyen material relevante que no se haya incluido en el cuerpo principal del documento, pero que contribuye significativamente a la comprensión y valoración del trabajo realizado.

21.1 Consulta de carga de datos

```
INSERT INTO wc_labeldataset (id, id_spot, spot, fecha, dir, velmedia, racha,
altura_oleaje, periodo_medio_oleaje,
periodo_pico_oleaje, direccion_oleaje, velocidad_viento, direccion_viento,
velocidad_corriente, direccion_corriente,
id_deporte, deporte, id_modalidad, modalidad, valoracion, perfil_rider, peso_rider,
edad_rider, kite_size, sail_size, wing_size, kite_board_size,
kite_board_type, wind_board_size, wind_board_type, wing_board_size, wing_ffoil_size,
labeled)
SELECT
  ROW_NUMBER() OVER () AS id,
  ws.id as id_spot,
  ws.nombre as spot,
  ph.fecha,
  ah.dir,
  round(ah.velmedia * 1.94384) as velmedia,
  round(ah.racha * 1.94384) as racha,
  ph.altura_oleaje,
  round(ph.periodo_medio_oleaje) as periodo_medio_oleaje,
  round(ph.periodo_pico_oleaje) as periodo_pico_oleaje,
  ph.direccion_oleaje,
  round(ph.velocidad_viento * 1.94384) as velocidad_viento,
  ph.direccion_viento,
  round(ph.velocidad_corriente) as velocidad_corriente,
  ph.direccion_corriente,
  wd.id as id_deporte,
  wd.deporte as deporte,
  wm.id as id_modalidad,
  wm.modalidad as modalidad,
  null as valoracion,
  null as perfil_rider,
  null as peso_rider,
  null as edad_rider,
  null as kite_size,
  null as sail_size,
  null as wing_size,
  null as kite_board_size,
  null as kite_board_type,
  null as wind_board_size,
  null as wind_board_type,
  null as wing_board_size,
  null as wing_ffoil_size,
  false as labeled
FROM
  wc_spot ws,
  wc_spot_puertos_punto_modelo wsppm,
  wc_spot_aemet_estacion wsae,
  aemet_historico ah,
  puertos_historico ph,
  wc_deporte wd,
  wc_modalidad wm
WHERE
  ws.id = wsppm.id_spot
  AND ws.id = wsae.id_spot
  AND date_trunc('day', ah.fecha) = date_trunc('day', ph.fecha)
  AND ah.indicativo = wsae.id_estacion
  AND ph.punto_modelo = wsppm.punto_modelo
  and ph.velocidad_viento < ah.racha --> La velocidad del viento no puede ser mayor a la
racha
  AND EXTRACT(HOUR FROM ph.fecha::timestamp) between 7 and 20;
```

21.2 Código función windcaddyalg

```
def windcaddyalg(peso, vel_viento, edad, racha, periodo):
    warnings.filterwarnings("ignore")
    val_data = np.array([peso, vel_viento, edad, racha,
periodo]).reshape(1, -1)
    val_model = loadmodel('grid_randomforest_valoracion.pkl')
    valoracion = val_model.predict(val_data)
    valoracion_value = obtener_valoracion_str(valoracion)
    if valoracion_value == "Ni me tiro":
        return "Ni me tiro", None, None

    sail_model = loadmodel('grid_xgbregressor_sail_size.pkl')
    sail_data = np.hstack((val_data, valoracion.reshape(-1, 1)))
    sail_size = sail_model.predict(sail_data)

    wbs_model = loadmodel('grid_randomforest_wind_board_size.pkl')
    wbs_data = np.hstack((np.hstack((val_data, sail_size.reshape(-1,
1))), valoracion.reshape(-1, 1)))
    wbs_size = wbs_model.predict(wbs_data)

    sail_size_value = sail_size[0].round(decimals=1)
    wbs_value = wbs_size[0].round(decimals=0)

    return valoracion_value, sail_size_value, wbs_value
```

21.3 Consulta de extracción de datos para validación visual

```
select * from
((SELECT CONCAT_WS(', ', '[', peso_rider, velocidad_viento, edad_rider, racha,
periodo_medio_oleaje, ']') as INPUT_VECTOR, CONCAT('#', valoracion, '
        ', sail_size,'            ', wind_board_size)
FROM wc_labeldataset_bck
WHERE valoracion = 1 ORDER BY RANDOM()
LIMIT 10)
union
(SELECT CONCAT_WS(', ', '[', peso_rider, velocidad_viento, edad_rider, racha,
periodo_medio_oleaje, ']') as INPUT_VECTOR, CONCAT('#', valoracion, '
        ', sail_size,'            ', wind_board_size)
FROM wc_labeldataset_bck
WHERE valoracion = 2 ORDER BY RANDOM()
LIMIT 10)
union
(SELECT CONCAT_WS(', ', '[', peso_rider, velocidad_viento, edad_rider, racha,
periodo_medio_oleaje, ']') as INPUT_VECTOR, CONCAT('#', valoracion, '
        ', sail_size,'            ', wind_board_size)
FROM wc_labeldataset_bck
WHERE valoracion = 3 ORDER BY RANDOM()
LIMIT 10)
union
(SELECT CONCAT_WS(', ', '[', peso_rider, velocidad_viento, edad_rider, racha,
periodo_medio_oleaje, ']') as INPUT_VECTOR, CONCAT('#', valoracion, '
        ', sail_size,'            ', wind_board_size)
FROM wc_labeldataset_bck
WHERE valoracion = 4 ORDER BY RANDOM()
LIMIT 10)
union
(SELECT CONCAT_WS(', ', '[', peso_rider, velocidad_viento, edad_rider, racha,
periodo_medio_oleaje, ']') as INPUT_VECTOR, CONCAT('#', valoracion, '
        ', sail_size,'            ', wind_board_size)
FROM wc_labeldataset_bck
WHERE valoracion = 5 ORDER BY RANDOM()
LIMIT 10))
order by 2
```

21.4 Relación de los paquetes Python usados

Paquete	Version
absl-py	2.1.0
astunparse	1.6.3
certifi	2024.6.2
charset-normalizer	3.3.2
contourpy	1.2.1
cycler	0.12.1
debugpy	1.8.1
flatbuffers	24.3.25
fonttools	4.53.0
gast	0.5.4
google-pasta	0.2.0
grpcio	1.64.1
h5py	3.11.0
idna	3.7
joblib	1.4.2
keras	3.3.3
kiwisolver	1.4.5
libclang	18.1.1
Markdown	3.6
markdown-it-py	3.0.0
MarkupSafe	2.1.5
matplotlib	3.9.0
mdurl	0.1.2
ml-dtypes	0.3.2
namex	0.0.8
numpy	1.26.4
opt-einsum	3.3.0
optree	0.11.0
packaging	24.0

Paquete	Version
pandas	2.2.2
pillow	10.3.0
pip	24.0
protobuf	4.25.3
psycopg2	2.9.9
Pygments	2.18.0
pyparsing	3.1.2
python-dateutil	2.9.0.post0
pytz	2024.1
requests	2.32.3
rich	13.7.1
scikit-learn	1.5.0
scipy	1.13.1
seaborn	0.13.2
setuptools	70.0.0
six	1.16.0
tensorboard	2.16.2
tensorboard-data-server	0.7.2
tensorflow	2.16.1
tensorflow-intel	2.16.1
termcolor	2.4.0
threadpoolctl	3.5.0
typing_extensions	4.12.1
tzdata	2024.1
urllib3	2.2.1
Werkzeug	3.0.3
wheel	0.43.0
wrapt	1.16.0
xgboost	2.0.3

21.5 Ficheros Docker Formulario Etiquetado

A continuación, se comparten los archivos Docker a través de un enlace a OneDrive.

Cabe mencionar que el acceso está restringido a los usuarios del centro y deberán acceder con su cuenta de la UOC.

Ficheros Docker Formulario Etiquetado