Universidad Internacional de La Rioja

Escuela Superior de Ingeniería y Tecnología

Máster Universitario en Análisis y Visualización de Datos Masivos

Predicción del tiempo de corredores en carreras de ultradistancia

Trabajo Fin de Máster

Tipo de trabajo: Piloto experimental

Presentado por: Sayols Puig, Sergi

Director/a: Monsalve Torra, Dr. Ana Eddy

Resumen

Las carreras de montaña de ultradistancia transcurren por entornos remotos y se necesitan varias horas para completarlas. Esto representa un reto para los organizadores que deben garantizar la seguridad de la carrera. El *Centre Excursionista Torelló* organiza la carrera *Pels Camins dels Matxos*, y quiere incorporar una herramienta para estimar el tiempo en que un corredor debería pasar por un control y con esto anticiparse a posibles accidentes. Los modelos de regresión lineal múltiple se usan habitualmente para predecir el tiempo que necesitará un corredor para finalizar una carrera. El inconveniente de este método es que presupone un desarrollo lineal de la carrera y no tiene en cuenta cómo cambia el rendimiento del corredor a lo largo de la prueba. Otro problema habitual es no disponer de las variables predictoras más importantes ya que suelen ser parámetros antropométricos medidos en el laboratorio. Para abordar estos inconvenientes este trabajo de investigación propone probar dos tipos de algoritmos de *machine learning*, modelos de regresión no lineal, en concreto *Random Forests* (RF) y *Perceptrón Multicapa* (MLP), y entrenarlos utilizando los tiempos de los controles intermedios, la edad y el sexo de los corredores. Los algoritmos propuestos son comparados mediante el error medio absoluto (MAE) en la predicción de la llegada desde un control intermedio. Los resultados muestran que el MLP mejora en un 56% la precisión de RF (MAE MLP 9,8 minutos; MAE RF 22,4 minutos). Sin embargo, la tendencia del MAE en el MLP es aleatoria y no correlaciona con el número de tiempos conocidos usados en la predicción. El modelo basado en RF se ha integrado con la aplicación actual del Centro de Control de Carrera.

Palabras Clave: carrera, ultradistancia, predicción de tiempos, Random Forests, Perceptrón Multicapa.

Abstract

Ultra-trail races take place in remote places and runners need several hours to complete. This represents a challenge for organizers, who need to guarantee the safety of the race. The *Centre Excursionista Torello* organizes the ultra-trail race named *Pels Camins dels Matxos* and wants to incorporate a tool to estimate the time a runner should pass through an intermediate checkpoint, and with this prevent possible accidents. Multiple linear regression models are commonly used to predict the time needed for a runner to finish a race. The inconvenient of this method is that assumes runners' performance is constant along the race. Another issue is the difficulty to obtain the most important predictor variables, for they commonly are anthropometric parameters measured in the lab. To address these issues, this study proposes the use of two machine learning algorithms to solve the problem of estimating race times as a non-linear regression problem, trained using checkpoint times, sex and age of the participants. Specifically, Random Forests (RF) and Multilayer Perceptron (MLP). The precision of the algorithms was evaluated using the Mean Absolute Error (MAE) when estimating the time at arrival from an intermediate checkpoint. The results show that MLP improves the precision of RF by 56% (MAE MLP 9.8 minutes; MAE RF 22.4 minutes). However, the MAE in the MLP shows a random trend and does not correlate with the number of known checkpoints used in the prediction. The model based on RF was integrated within the software stack of the Race Control Centre.

Keywords: race, ultra-trail, time prediction, Random Forests, Multilayer Perceptron.

Índice de contenidos

Índice de tablas

Índice de figuras

1. Introducción

1.1. Justificación

Las carreras de montaña se han popularizado en los últimos años y atraen, además de atletas profesionales, a corredores populares y amateurs. Según la Asociación Internacional de *Trail Running* (ITRA por sus siglas en inglés), que gestiona el circuito oficial de carreras de montaña a nivel mundial, existen 1,7 millones de corredores afiliados y 6.330 carreras oficiales en los 5 continentes programadas para el año 2021 (ITRA, 2021). Solo en España hay listadas 119 carreras, incluyendo las conocidas Transvulcania, con un recorrido que transcurre por las islas Canarias; la *Ultratrail* de los Pirineos, de 100km de distancia; o la *Zegama-Aizkorri*, una maratón de montaña en el País Vasco. Una parte importante de las carreras son de media o larga distancia, e implican distancias superiores a las de maratón (42km) y pueden tomar hasta 24 horas o más en completarse. Acostumbran a transcurrir parcialmente por entornos naturales remotos, alejados de pueblos y ciudades, y a menudo por alta montaña.

Las largas distancias y la belleza de los entornos por los que transcurren son los principales estímulos para la participación en este tipo de carreras. Sin embargo, estas mismas razones suponen un reto importante para los corredores y los organizadores. Estos últimos deben aplicar las medidas adecuadas para garantizar el desarrollo de la carrera en un entorno seguro y evitar accidentes, ya sean abandonos, lesiones o la pérdida de los corredores por el recorrido. Por este motivo es importante disponer de todas las herramientas necesarias que permitan anticiparse a eventos adversos, siendo una de ellas la estimación del tiempo en que un corredor debería pasar por un control.

El *Centre Excursionista Torelló* organiza la carrera de montaña de ultradistancia *Pels Camins dels Matxos* (Centre Excursionista Torelló, 2021), de 63km de distancia y que requiere de entre 6 y 15 horas para completar. Para la próxima edición quiere incorporar una herramienta para estimar el tiempo en que un corredor debería pasar por un control, e integrarla con la aplicación actual del Centro de Control de Carrera. De esta manera pretende adelantarse a sucesos adversos y así aumentar la seguridad de los corredores.

Existen actualmente unas pocas soluciones comerciales (*ChampionChip*, 2021; *LiveTrail*, 2021) y metodologías descritas en trabajos académicos (Fogliato et al., 2020, Knechtle et al., 2014) que desarrollan soluciones al problema de la predicción de tiempos. Sin embargo, todas ellas tienen desventajas evidentes como el precio, la imposibilidad de obtener todos los datos de entrenamiento que necesita el modelo propuesto, o la dificultad de integrarse con la plataforma tecnológica existente.

1.2. Planteamiento del trabajo

Gracias a las mejoras en las comunicaciones inalámbricas y la reducción del precio de la tecnología, actualmente es posible hacer un seguimiento casi en tiempo real de los corredores. En combinación con datos de ediciones anteriores, así como las características del terreno, meteorológicas y de condiciones de carrera, y usando técnicas de aprendizaje automático (*machine learning)* se puede hacer una predicción de cuándo los corredores van a pasar por los próximos controles de carrera.

El objetivo de este trabajo será desarrollar un modelo predictivo utilizando técnicas de *machine learning* modernas utilizando los datos de ediciones anteriores de la carrera *Pels Camins dels Matxos* e integrarlo con la plataforma tecnológica del Centro de Control de Carrera.

Se plantea: i) explorar diferentes modelos predictivos entrenados a partir de datos de ediciones anteriores para la carrera organizada por el Centre Excursionista Torelló; ii) desplegar el mejor modelo en producción, encapsulado en una API que pueda ser accedida por los demás microservicios que forman la actual aplicación del Centro de Control de Carrera; iii) desplegar la infraestructura de comunicación entre microservicios de la aplicación, utilizando un *bróker* de mensajes para el *streaming* de datos.

1.3. Estructura de la memoria

Este capítulo 1 es una introducción a las carreras de montaña en general, y en particular a *Pels Camins dels Maxtos*. Resume los objetivos que se buscan con el desarrollo de este trabajo, y el planteamiento de la resolución en sí.

El capítulo 2 describe las soluciones más prominentes que se utilizan actualmente en la predicción de carreras de montaña. Repasa algunos sistemas comerciales, *frameworks open-source* y hace un recorrido por la literatura disponible sobre la predicción de tiempos en carreras atléticas a pie. Se identifican los principales métodos, sus ventajas e inconvenientes, y los datos que son más importantes para entrenar los modelos predictivos. En menor medida, se discuten algunas de las herramientas que permiten entrenar y desplegar modelos predictivos, y la interacción con el resto de *los microservicios* de la aplicación y con el usuario final.

El capítulo 3 describe de manera más específica el objetivo de este trabajo, aportando mayor detalle al problema de la predicción de tiempos y los desafíos que hay que abordar para incorporarlo a la plataforma tecnológica del Centre Excursionista Torelló. Se definen los

algoritmos de *machine learning* que se utilizarán para predecir los tiempos, qué lenguaje/*framework* se va a utilizar, cómo se va a desplegar el modelo a través de una web *Application Programming Interface* (API), y finalmente cómo se conecta el modelo con el resto de la aplicación (bases de datos y microservicios). Se describe finalmente la metodología seguida para alcanzar los objetivos planteados.

El capítulo 4 se describe el desarrollo específico de la contribución identificando cuáles son los algoritmos que mejor se adaptan a nuestros datos, evaluando los resultados de la predicción utilizando datos de validación a través de métricas objetivas. También se muestra el resultado final de la incorporación del nuevo microservicio de predicción dentro de la aplicación actual del Centro de Control de Carrera. La última parte de este capítulo está dedicada a discutir la relevancia y peculiaridades de los resultados.

Finalmente, en el capítulo 5 se recogen las conclusiones de este trabajo, resumiendo cuál ha sido el algoritmo más adecuado para implementar el predictor, y qué nuevas partes se han implementado en la aplicación del Centro de Control de Carrera. Se describen las limitaciones del modelo actual y las posibles mejoras que podrían implementarse en el futuro.

2. Contexto y estado del arte

2.1. Resumen de la carrera *Pels Camins dels Matxos*

Pels Camins dels Matxos es el nombre de la carrera de *ultradistancia* organizada por el *Centre Excursionista Torelló*. Es una carrera muy popular dentro del circuito de carreras de montaña que se organizan en Cataluña, y es técnicamente demandante. Discurre durante 63 kilómetros por los espacios naturales del Collsacabra y la Serra de Guilleries (Figura 1), con un desnivel positivo acumulado de 6.200 metros. La carrera empieza y acaba en la localidad de Torelló, provincia de Barcelona. El punto más alto es la cumbre del Puigsacalm, de 1.512 metros de altura y una de las montañas emblemáticas dentro del excursionismo catalán (Figura 2).

Figura 1: Recorrido de la carrera Pels Camins dels Matxos, puntos de control y distancia recorrida. Fuente: Centre Excursionista Torelló (https://www.matxos.cat/)

La carrera se celebra anualmente. La primera edición ocurrió en el año 2001, donde concentró unas pocas decenas de corredores, hasta la última edición del año 2019 con un aforo limitado

a un máximo de 1.000 corredores. La única edición en que no se celebró fue en 2020 por la pandemia de COVID-19, que obligó primero a posponer la carrera, y finalmente a suspenderla. El recorrido ha ido cambiando hasta llegar al trazado actual en el año 2014. La carrera puntúa para el circuito internacional de carreras de montaña y sirve como clasificatoria para carreras de primera categoría, como la conocida *Ultra Trail du Montblanc*. Por este motivo atrae a muchos corredores y tiene limitado el aforo a un máximo de 1.000 participantes. Entre ellos se encuentran atletas profesionales, pero principalmente muchos amateurs. Los primeros corredores finalizan en un tiempo de 6 horas aproximadamente, mientras que los últimos en finalizar lo hacen en 15 horas (límite fijado por la organización). Según los organizadores, el porcentaje de corredores que no finalizan la carrera se sitúa en torno al 30% (Centre Excursionista Torelló, 2021).

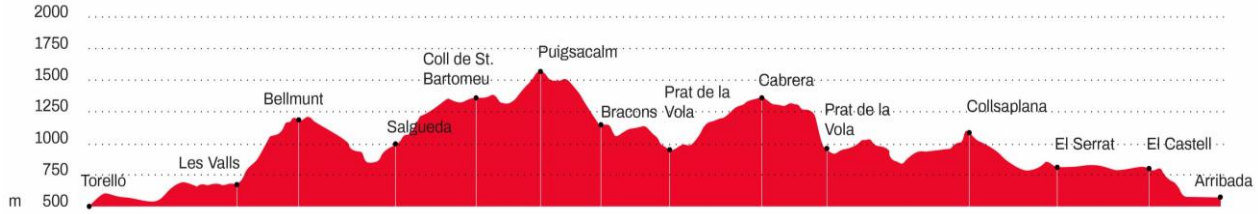

Figura 2: Perfil de altura de la carrera Pels Camins dels Matxos. Fuente: Centre Excursionista Torelló (https://www.matxos.cat/)

La actual aplicación del Centro de Control de Carrera recoge los tiempos de los corredores cuando pasan por uno de los controles de tiempo de la carrera. Estos son transmitidos desde el punto de control hacia el Centro de Control de Carrera a través de conexiones de datos móviles GSM donde hay cobertura móvil 3G, o alternativamente se usan antenas dirigidas *punto-a-punto*. Una vez llegados, se validan y almacenan en una base de datos SQL que el público general puede consultar a través de una aplicación web (Figura 3).

Actualmente la aplicación no dispone de la funcionalidad de estimar la hora en la que los corredores pasarán por los puntos de control. Tal y como recoge el recuadro rojo en la Figura 3, la nueva funcionalidad a implementar quedaría integrada dentro de la aplicación del Centro de Control de Carrera como un *microservicio* aislado que se comunica con el resto a través del paso de mensajes.

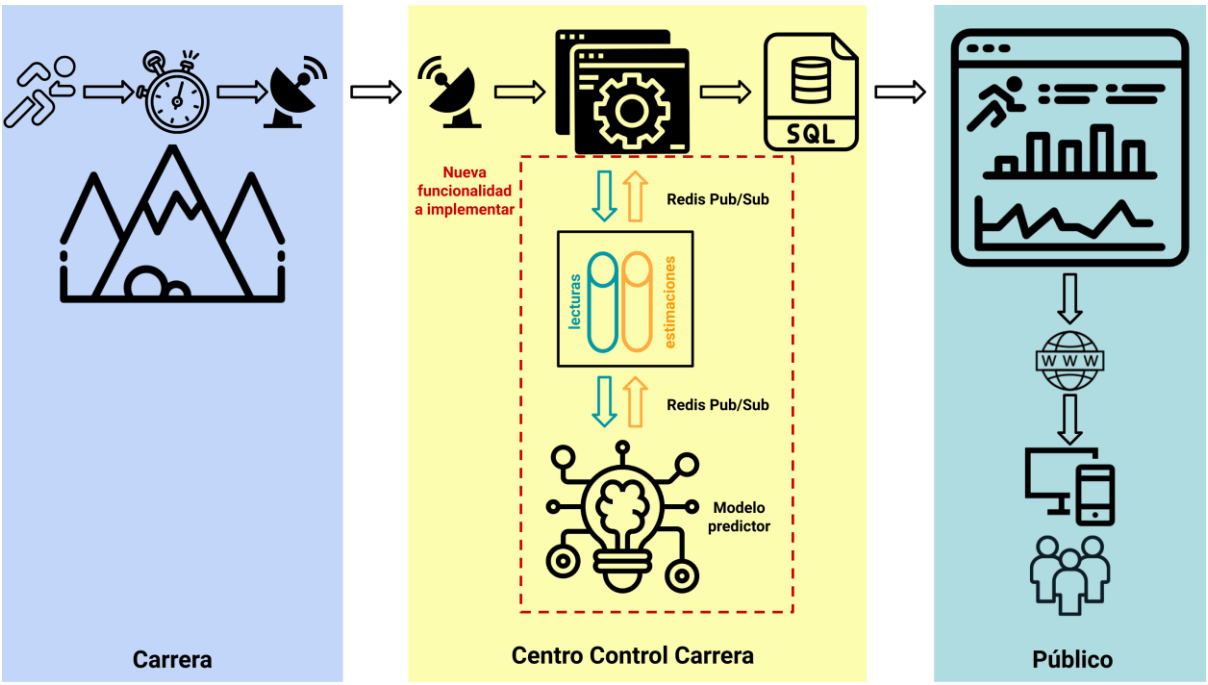

Figura 3: Arquitectura de la aplicación del Centro de Control de Carrera. Fuente: elaboración propia.

2.2. Estado del arte y trabajos relacionados

Existen numerosos trabajos científicos publicados que estudian la predicción del rendimiento de los atletas. La siguiente pretende ser una recopilación no exhaustiva pero representativa de la literatura actual sobre el tema.

El trabajo realizado por Knechtle et al. (2014) estudió la predicción de tiempo de carrera de medio maratón en corredores recreativos femeninos y masculinos. Su objetivo fue mejorar las ecuaciones utilizadas hasta el momento para predecir los tiempos de medio maratón, utilizando como variables predictoras el porcentaje de grasa corporal y la velocidad durante las sesiones de entrenamiento. Para ello, estudiaron una muestra aleatoria simple de 147 hombres y 83 mujeres para determinar los coeficientes de un modelo lineal de regresión múltiple. Comprobaron que los tiempos estimados por la ecuación de su modelo correlacionaban de manera significativa con los tiempos observados (hombres: R^2=0.71, p-valor<0.0001; mujeres: R^2=0.89, p-valor<0.0001; donde R^2 es el porcentaje de variación de la variable dependiente que es explicado por su relación con las variables predictoras).

El trabajo realizado por Alvero-Cruz et al. (2019) estudió la predicción del rendimiento en carreras de montaña de corta distancia utilizando únicamente variables antropométricas. A partir de una muestra de 11 hombres de edades entre 30 y 40 años, recopilaron los tiempos que tomaron en finalizar una carrera de montaña de 27 kilómetros de distancia. Utilizaron

estos tiempos para estimar los coeficientes de un modelo lineal de regresión múltiple donde la variable dependiente era el tiempo de carrera. Como variables independientes (predictoras) utilizaron distintas variables antropométricas candidatas a tener un efecto sobre la variable dependiente. Éstas fueron la edad, los años de entrenamiento del corredor, el índice de masa corporal (BMI), la capacidad pulmonar (VO_2 max), las pulsaciones por minuto máximas y el nivel de lactato en sangre. Concluyeron que las únicas variables asociadas significativamente con el tiempo de carrera fueron BMI y VO_2 max. Concluyeron que el modelo lineal que mejor predecía el tiempo de carrera utilizaba únicamente estas 2 variables como coeficientes, según la siguiente fórmula: $tiempo\ (min) = 203.9956 - 1.9001 \times VO2\ max + 10.2816 \times BMI\%$. Los tiempos estimados correlacionaban de manera significativa con los tiempos observados (R^2=0.839, p-valor=0.0007).

Keogh et al. (2019) hacen una revisión exhaustiva de la literatura científica buscando qué ecuaciones se utilizan comúnmente en la predicción de tiempos en carreras de maratón. Encontraron 36 estudios con 114 ecuaciones distintas. Todas ellas eran modelos lineales de regresión múltiple de la forma $y=\beta_0+\beta_1 x_1+\beta_2 x_2+...+\beta_p x_p+\epsilon$, y se diferenciaban por utilizar una combinación distinta de variables predictoras. Encontraron que 61 de los estudios analizados utilizaban parámetros antropométricos del corredor como variables predictoras, mientras que 53 de ellos utilizaban parámetros los cuales era necesario medirlos utilizando material de laboratorio. La precisión de esas ecuaciones se media calculando la correlación del tiempo estimado con el tiempo real. Los valores de correlación obtenidos oscilaban entre R^2=0.1 y R^2=0.99.

Otros estudios han concluido que hay variables que son especialmente relevantes en la predicción de tiempos en carreras. Cuk et al. (2020) estudió 4807 hombres y 1278 mujeres corredores de maratón para concluir que, aunque los hombres son significativamente más rápidos en completar la carrera, las mujeres son más constantes durante la carrera (variación absoluta media de la velocidad en hombres=5.46%, mujeres=4.12%). Coast et al. (2004) comparó los mejores tiempos de hombres y mujeres en distancias de 100 metros hasta 200 kilómetros, concluyendo que los hombres son de media un 12.4% más rápidos. Hubble & Zhao (2016) estudiaron la influencia de seguir una estrategia subóptima durante la carrera, y concluyeron que los hombres siguen de manera más frecuente una estrategia errónea que les ralentiza en los últimos kilómetros. Nikolaidis & Knechtle (2017) estudiaron los tiempos de los participantes de la maratón de Nueva York entre los años 2006 y 2016, y encontraron una diferencia significativa en la capacidad de los atletas de mantener el mismo ritmo a lo largo de la carrera según el grupo de edad al que pertenecían (p-valor < 0.001, rango variación media de la velocidad [0.001, 0.004]). Rüst et al. (2011) estudió el rendimiento de 84 hombres en

una carrera de medio maratón y concluyó que la variable más importante para la estimación del tiempo de llegada era el BMI (R^2=0.56, p-valor=0.0150) seguido por la velocidad en los entrenamientos (R^2=-0.54, p-valor=0.0045). En un estudio parecido realizado por Knechtle et al. (2011) con 42 participantes mujeres en la media maratón de Basel, determinaron que el tamaño medio en milímetros del pliegue cutáneo a la altura de la axila junto con la velocidad en los entrenamientos son las variables más relevantes para la predicción (R^2=0.71 entre el tiempo estimado por regresión lineal y el tiempo real). Finalmente, Ely et al. (2007) estudió los datos meteorológicos y resultados de las maratones de Boston, Nueva York, Twin Cities, Grandma's, Richmond, Hartford y Vancouver, en un periodo de entre 10 a 36 años, y concluyó que hay una progresiva ralentización de los corredores cuando la temperatura WBGT aumenta de los 5ºC a los 25ºC (WBGT es *wet-bulb globe temperature* por sus siglas en inglés, y es una medida combinada de temperatura, humedad, velocidad del viento, ángulo de incidencia del sol e índice de irradiación solar).

Específicamente para carreras de montaña, Fogliato et al., (2020) proponen una metodología estadística para calcular la probabilidad de un corredor de llegar al siguiente control, y estimar el tiempo necesario para hacerlo. Para la estimación de los tiempos, proponen 4 algoritmos distintos:

- Un modelo de regresión lineal, utilizado como *baseline* con el que comparar los demás métodos.
- Un modelo de regresión lineal con regularización LASSO (Tibshirani, 1996). LASSO es un método para corregir el sobreajuste, penalizando las variables menos importantes del modelo.
- *Random Forests* (Breiman, 2001), que puede funcionar bien en el caso que haya relaciones no lineales entre el tiempo y las variables predictoras. El algoritmo de se explica en detalle en la sección específica *4.3.1. Random Forests* durante el desarrollo de la contribución.
- *Gradient Boosted Trees* (Chen & Guestrin, 2016), un método que igual que *Random Forests* combina múltiples árboles de decisión.

Los autores del estudio utilizaron los tiempos de los corredores que participaron en la carrera del *Ultra Trail del Montblanc* para comprobar cuál de los modelos proporcionaba un error más bajo en la predicción. Concluyeron que el modelo de regresión lineal simple proporcionaba un mayor error en la predicción (raíz del error medio cuadrado, RMSE) mientras que los modelos no lineales basados en árboles proporcionaban los errores menores (Figura 4).

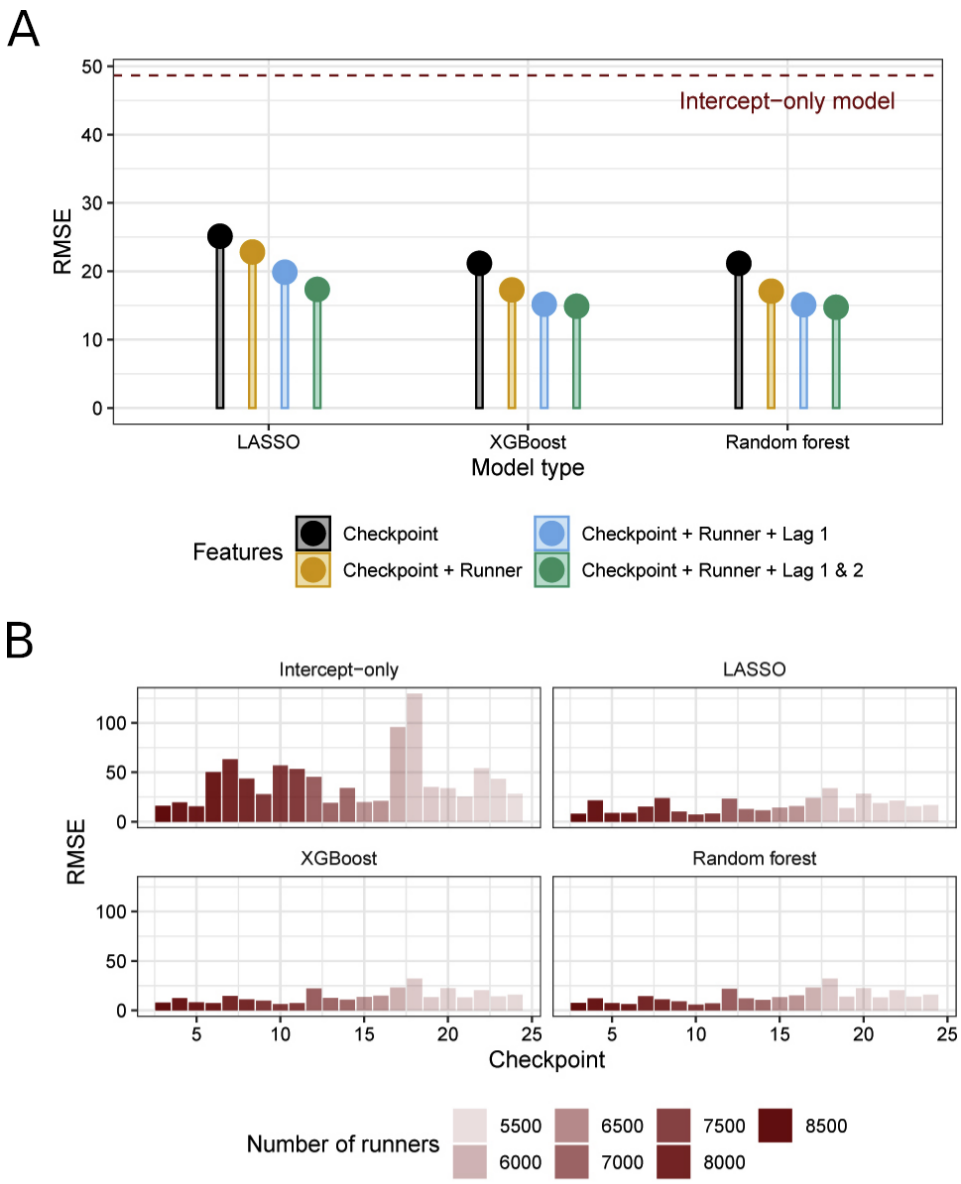

Figura 4: Error de la predicción según el método desarrollado por TRAP. A: RMSE utilizando diferentes variables predictoras. B: RMSE para cada control. Fuente: Fogliato et al., 2020.

Finalmente, algunos proveedores de tecnología de cronometraje ofrecen, bajo pago, la predicción de tiempos como parte del paquete de servicios que ofrecen (ChampionChip, 2021; LiveTrail, 2021). Desafortunadamente, estas dos empresas no aportan información sobre cómo está implementado su modelo predictor ni qué algoritmos utilizan. Además, las soluciones comerciales son caras y destinadas a las carreras más profesionales que mueven una mayor cantidad de dinero. Sin embargo, se asume que las predicciones que ofrecen estos servicios son precisas y mejoran año tras año ya que las empresas tienen un incentivo económico para hacerlo.

2.3. Conclusiones

Las carreras atléticas de maratón o medio maratón que discurren por circuitos llanos son las más estudiadas. Por un lado, son más populares que las carreras de montaña, y por otro, su popularización llegó mucho antes. Dadas las características homogéneas del terreno y la distancia de estas carreras, se emplean habitualmente modelos lineales para la predicción de los tiempos de paso. Por el contrario, las carreras de montaña discurren por terrenos desiguales con grandes pendientes y los modelos no lineales consiguen una mayor precisión.

Las características antropométricas de los corredores son utilizadas frecuentemente como variables predictoras para el rendimiento de los corredores. Son las más relevantes para realizar la predicción, junto con la velocidad en los entrenamientos. Desafortunadamente, aparte del sexo y la edad, el resto de las variables descritas en los trabajos estudiados son difíciles de conseguir sistemáticamente.

Revisada la literatura, se ha comprobado que los modelos estudiados utilizan como variables predictoras datos biométricos de los que no disponemos. Su integración en la actual aplicación del Centro de Control de Carrera sería muy complicada. También existen empresas especializadas en la organización de carreras que ofrecen bajo pago un servicio de predicción de tiempos. Sin embargo, no se dispone de información sobre cómo están construidos los modelos ni cuál es su precisión.

Por consiguiente, se puede concluir que será necesario implementar un modelo de predicción a medida para la carrera *Pels Camins dels* Matxos, utilizando técnicas modernas de *machine learning* y solamente las variables predictoras de las que disponemos para esta carrera. Así se implementaría una solución de calidad, económica y fácilmente integrable con la aplicación del Centro de Control de Carrera.

3. Objetivos concretos y metodología de trabajo

3.1. Objetivo general

El objetivo general de este trabajo consiste en desarrollar un prototipo de *software* que permita predecir los tiempos de paso de los corredores por los diferentes controles de la carrera *Pels Camins dels Matxos* (PCDM). Para ello se utilizarán los datos de las ediciones anteriores de la carrera y se estudiarán 2 tipos de algoritmos de *machine learning* para integrar el que mejor precisión tenga dentro de la actual aplicación del Centro de Control de Carrera (CCC).

3.2. Objetivos específicos

- Integrar los datos de las distintas ediciones de la carrera, y describir las variables más importantes.
- Determinar la capacidad de los algoritmos de *Random Forests* y del *Perceptrón Multicapa* para predecir los tiempos de llegada a partir de datos de ediciones anteriores.
- Encapsular el modelo de predicción más preciso en un microservicio web.
- Implementar un *bróker* de mensajes para facilitar la comunicación entre los demás microservicios de la aplicación del CCC y la API de predicción.

3.3. Metodología del trabajo

Este trabajo sigue una metodología similar a la de otros proyectos de *machine learning* (Figura 5) para desplegar en el entorno de producción un modelo predictivo construido sobre un conjunto de datos.

Consta de tres fases bien diferenciadas:

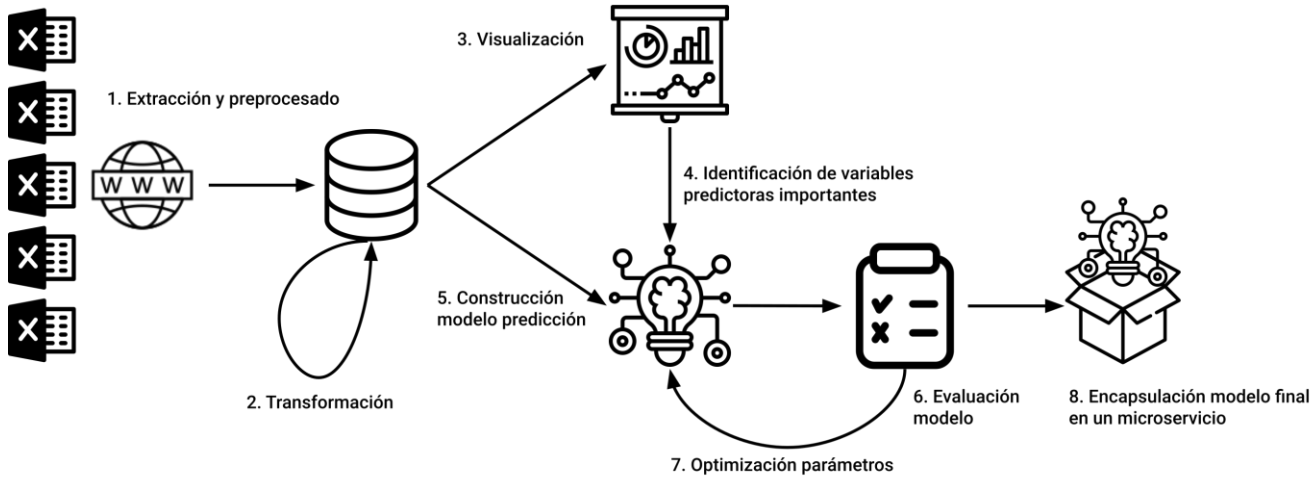

Figura 5: metodología de trabajo. Fuente: elaboración propia.

1. Extracción y preparación de los datos

    - Extracción: obtención de los datos de las ediciones anteriores de la carrera. Son públicos desde el año 2014 y se pueden descargar de la web del *Centre Excursionista Torelló* (https://www.matxos.cat/). Estos se van a complementar con datos internos no públicos: el sexo del corredor y la fecha de nacimiento.

    - Preprocesado: las tablas de datos obtenidas en el proceso anterior se van a normalizar de manera que todas las tablas de datos contengan las mismas columnas, en el mismo orden, utilizando los mismos separadores entre campos. Los datos de las distintas ediciones se van a agregar y almacenar en una base de datos central.

    - Transformación: los datos de la fase anterior son parcialmente incompletos y será necesario transformarlos antes de trabajar con ellos. Habrá que realizar un proceso para detectar registros incompletos. Mediante el uso de técnicas estadísticas se imputarán los tiempos que falten, en concreto se utilizará *Random Forests* para construir un modelo que impute mediante la proximidad de otros corredores. Otros campos se imputarán mediante otras técnicas, en función de su tipología. Estos se detallarán más adelante en el capítulo 4.

2. Construcción y evaluación del modelo predictor

    - Visualización e identificación de las variables predictoras importantes: A partir de la base de datos construida en la fase anterior, se van a realizar las

visualizaciones necesarias para hacer una exploración del conjunto de datos de entrada. Para ello se va a utilizar R. El objetivo de estas visualizaciones es entender la distribución de los datos y tener una idea de la importancia de cada variable en la elaboración del modelo de predicción.

- Construcción y evaluación: se utilizarán las técnicas *Random Forests* (RF) y *Redes Neuronales* (NN) en una conformación de *Perceptrón Multicapa* (MLP) para entrenar dos modelos predictores. Puesto que se trata de modelos de regresión, se evaluarán utilizando como métrica el *Mean Absolute Error (MAE)*. Este expresa el error medio del modelo en unidades de la variable objetivo de la predicción (segundos, en este caso), facilitando su interpretación.

3. Puesta a producción: encapsulamiento del mejor modelo en un microservicio, accesible a través de una web API.

4. Desarrollo específico de la contribución

4.1. Extracción y transformación de datos

El proceso completo de extracción y transformación de datos previo a realizar los modelos se resume en la siguiente Figura 6:

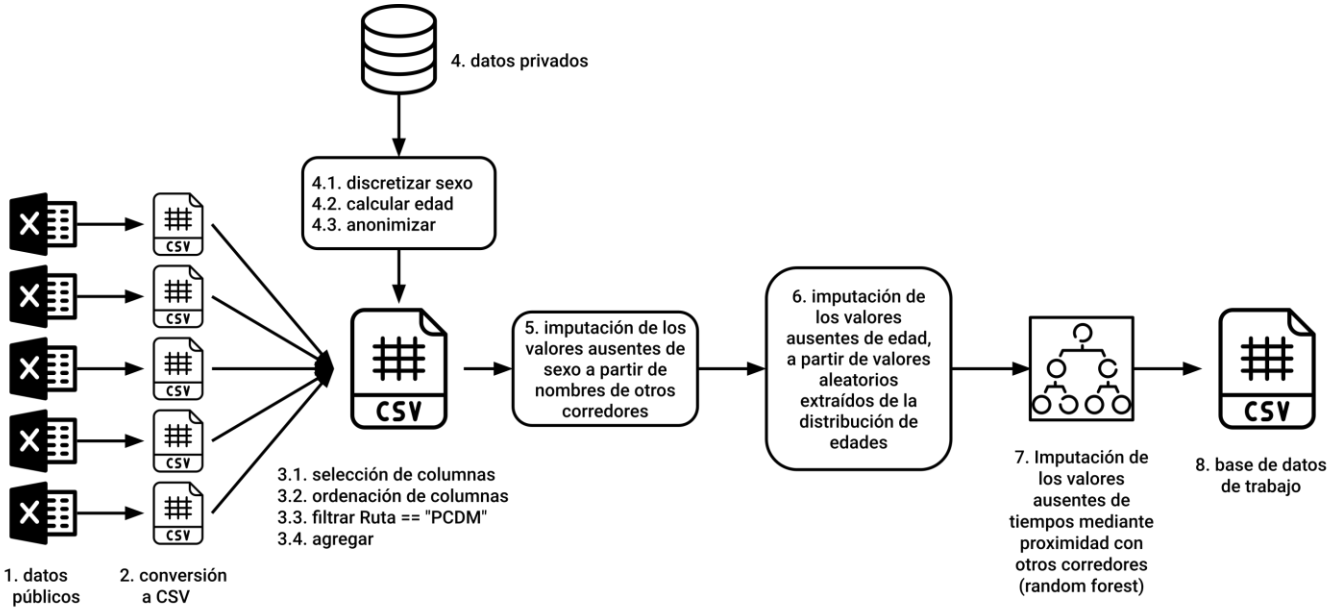

Figura 6: Resumen del proceso de extracción y transformación de datos seguido. Fuente: elaboración propia.

Se han obtenido los datos de las ediciones anteriores (2014-2019) de la carrera de la web del *Centre Excursionista Torelló* (www.matxos.cat). Estos están disponibles como hojas de cálculo Excel, con un fichero para cada edición. El formato de estas hojas es consistente y contiene una entrada para cada corredor (filas) y una columna para el tiempo registrado en cada control además del nombre del corredor, año de la edición, número de dorsal y recorrido (Tabla 1).

Las columnas con los controles están ordenadas por distancia respecto la salida (Tabla 10). El número que contienen representa el número de segundos transcurridos desde la salida.

En estas tablas hay resumidos los tiempos de varios recorridos. Efectivamente son carreras distintas, y solo estamos interesados en estimar los tiempos para el recorrido de PCDM. Para ello, se han filtrado todas las filas donde la columna *Ruta* contiene un valor distinto a PCDM.

Para agregar las hojas de las distintas ediciones, primero se han convertido a formato texto separado por comas (CSV). Aunque en principio el número de columnas y su orden es consistente en todas las hojas, para hacer el proceso más robusto a posibles cambios en futuras ediciones, se ha realizado de manera explícita la selección y ordenación de las

columnas en un orden concreto. Finalmente, los datos son agregados en un único archivo CSV con todos los registros de todas las ediciones.

Este archivo CSV se ha complementado con datos internos no públicos que el corredor proporciona en el proceso de inscripción. Estos serían los datos del sexo y la fecha de nacimiento del corredor. El sexo se ha discretizado en un rango de enteros [0, 1], siendo los hombres identificados por el número 0, y las mujeres por un 1. A partir de la fecha de nacimiento y la edición, se ha calculado la edad del corredor en el momento que realizó la carrera.

Los registros contienen datos ausentes (<5%). Los hay en los campos de sexo, edad y en todos los controles intermedios. Dada la diferente tipología de estos campos es necesario seguir estrategias distintas para imputarlos:

1. Para imputar los valores ausentes de sexo (29 de 1.721, 1,6%) se ha construido un diccionario de nombres→sexo con los demás corredores, y se han podido identificar de manera inequívoca todos los valores ausentes.

2. Para imputar los valores ausentes de edad (32 de 1.721, 1,8%) se han tomado de manera aleatoria de una distribución construida con las edades de todos los corredores, con probabilidad igual a sus frecuencias.

3. Para imputar los tiempos ausentes de los controles (1.074 de 17.857, 5,6%) se ha utilizado la función *rfImpute()* del paquete *randomForest* (Liaw & Wiener, 2002) para el software estadístico R (*R Core Team*, 2021). Esta función realiza la estimación de un valor continuo utilizando los valores conocidos de las instancias más próximas del conjunto de datos. De esta manera garantizamos que los tiempos imputados son parecidos a los de otros corredores que pasaron con tiempos similares en los demás controles.

Finalmente, los datos han sido anonimizados de manera que no se puedan relacionar las instancias con las personas reales que corrieron la carrera en ediciones anteriores. Para ello, se ha eliminado completamente la columna nombre del conjunto de datos.

La tabla final (Tabla 2) con los datos de entrenamiento contiene solamente las columnas con información del sexo, edad del corredor, y tiempos de paso intermedio. Está completada con los datos imputados, transformados y anonimizados.

Edicion	Dorsal	Nombre	Les Valls	Bellmunt	Salgueda	Sant Bartomeu	Puigsacalm	Prat de la vola 1	Cabrera	Prat de la vola 2	Collsaplana	Serrat	Llegada	Ruta
2014	49	Josep Cuadrat	1924		6513	7856	9197	11400	13560	15094	17409	20371	23100	PCDM
2014	613	Gerard Morales	1921	3819	6579	7867	9352	11410	13680	15222	17659	20668	23280	PCDM
2014	382	Ernest Ausiro	1943	3908	6613	7914		11460	13800	15338	17789	20791	23460	PCDM
2014	604	Xavi Padros	1940	4063	6747	8163	9653	11700	14100			21434	24000	PCDM
2014	467	Gabriel Crosas	1923	3815	6526	7854	9227	11403	13560	15092	17509	20845	24060	PCDM

Tabla 1: Ejemplo de un subconjunto de datos disponibles públicamente de las ediciones anteriores de la carrera. Fuente: www.matxos.cat

sexo	edad	Les Valls	Bellmunt	Salgueda	Sant Bartomeu	Puigsacalm	Prat de la vola 1	Cabrera	Prat de la vola 2	Collsaplana	Serrat	Llegada
H	35	1924	3838	6513	7856	9197	11400	13560	15094	17409	20371	23100
H	37	1921	3819	6579	7867	9352	11410	13680	15222	17659	20668	23280
H	33	1943	3908	6613	7914	9445	11460	13800	15338	17789	20791	23460
H	22	1940	4063	6747	8163	9653	11700	14100	15645	18081	21434	24000
H	25	1923	3815	6526	7854	9227	11403	13560	15092	17509	20845	24060

Tabla 2: Ejemplo del subconjunto de datos después de preprocesar y anonimizar. Fuente: elaboración propia.

4.2. Descripción del conjunto de datos

La carrera *Pels Camins dels Matxos* transcurre durante 63 kilómetros por un recorrido circular (Figura 1, Figura 7A). Consta de 12 controles cronometrados distribuidos a lo largo del recorrido, separados de media por 5,7 km (mínimo 3 km, máximo 9 km, mediana 5 km). El desnivel positivo entre controles también es variable (Figura 2). La carrera tiene varios recorridos de distinta distancia, siendo el más popular *Pels Camins dels Matxos*, en la que han participado 1.721 corredores desde la edición del 2014 (Figura 7B).

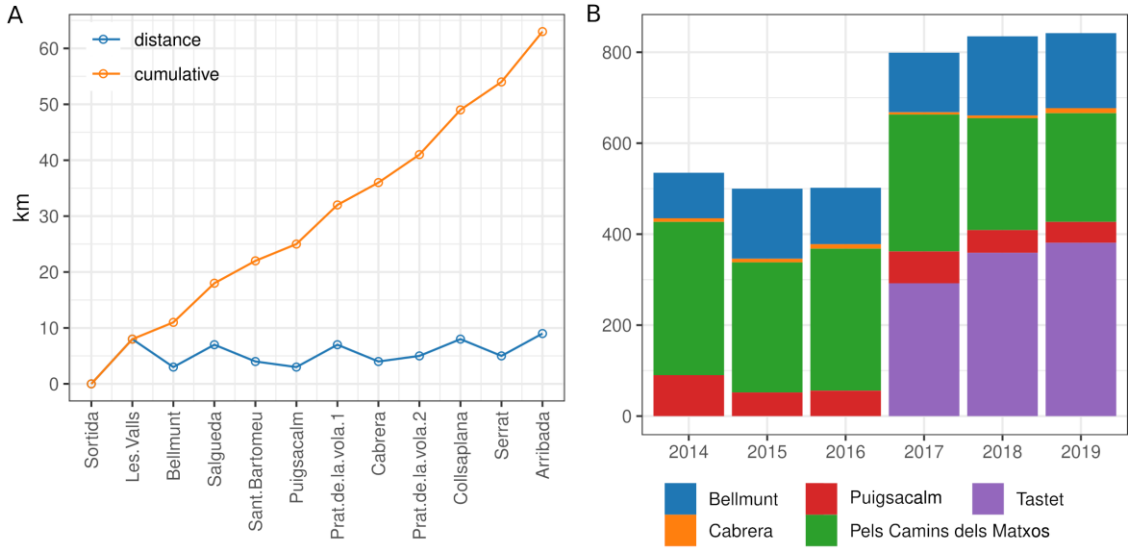

Figura 7: Descripción de los datos básicos de la carrera. A: distancia total (azul) y distancia acumulada (naranja) entre los distintos controles de la carrera; B: corredores de cada recorrido de la carrera. Fuente: elaboración propia.

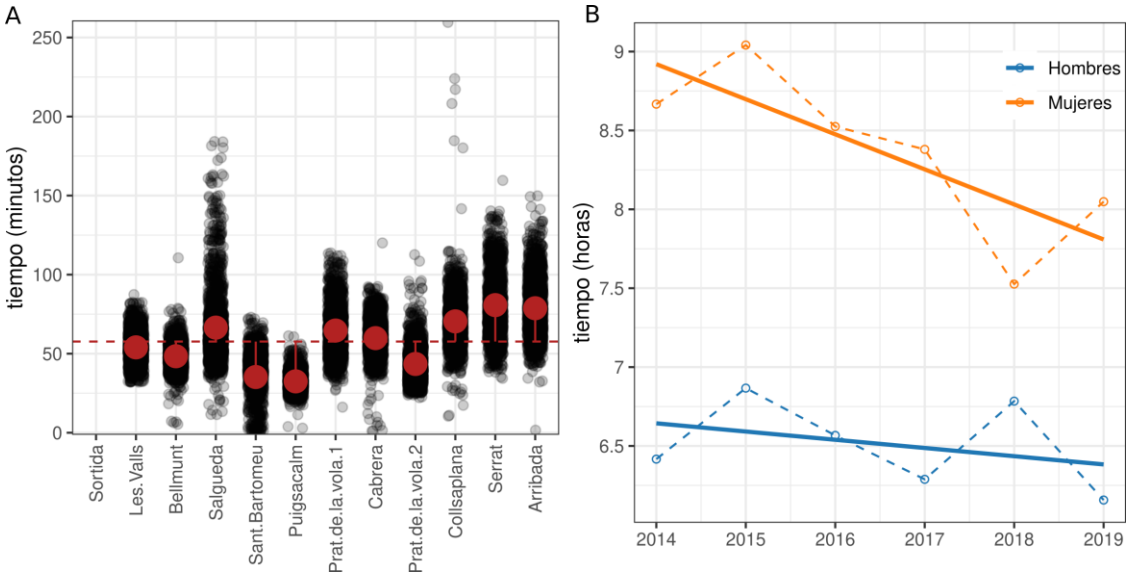

Figura 8: Descripción de los datos básicos de la carrera. A: tiempo (minutos) de cada corredor para completar un control; B: evolución del tiempo del ganador de la carrera, segmentado por sexo. Fuente: elaboración propia.

El tiempo que toman los participantes para llegar de un control a otro es variable. Éstos, necesitan de media 56 minutos para acceder al siguiente control tal y como indica la línea discontinua en la Figura 8A. El punto rojo muestra por cada control el tiempo medio necesario para completarlo, siendo distinto para cada uno de los controles. En los últimos controles del recorrido se aprecia que el tiempo medio evoluciona al alza, así como un aumento en la dispersión de tiempos.

La Figura 8B presenta la evolución del tiempo de llegada del mejor corredor desde la edición del año 2014. Se aprecia una tendencia hacia mejores tiempos en los hombres (representados en azul en la figura), mucho más prominente en el grupo de mujeres (naranja). La línea discontinua une los puntos con el mejor tiempo de cada edición, mientras que la línea continua muestra el ajuste del modelo lineal del mejor tiempo respecto la edición, donde la pendiente de la recta representa la tendencia.

De media, las mujeres necesitan 1 hora y 17 minutos más que los hombres en completar el recorrido (media hombres: 10 horas 23 minutos, mujeres: 11 horas 40 minutos; Figura 9A). En la Figura 9A, los extremos de la caja representan los cuartiles 25 y 75, la línea central el cuartil 50, y los extremos de los bigotes 1.5 veces el rango intercuartílico. Igualmente, el conjunto de las mujeres tiene sensiblemente una mayor dispersión que los hombres en lo que respecta al tiempo medio para completar un control (Figura 9B).

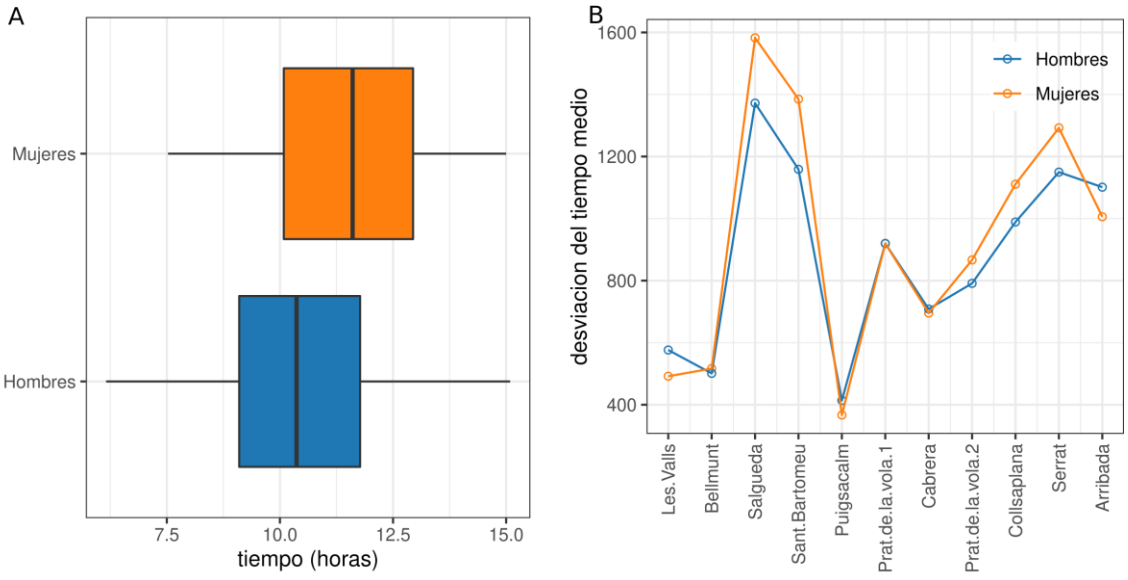

Figura 9: Diferencia de tiempo entre hombres y mujeres. A: tiempo necesario para terminar la carrera, segmentado por sexo (mujeres: naranja, hombres: azul); B: desviación absoluta en segundos sobre el tiempo medio necesario para completar el control. Fuente: elaboración propia.

El tiempo en completar la carrera mejora según la edad aumenta, hasta encontrar su mínimo a los 34,7 años de edad, tal y como muestra la flecha roja en la Figura 10. La línea azul muestra un ajuste de los datos mediante el método de regresión polinómica local *loess* (Cleveland, 1979), y la sombra gris el error estándar del ajuste. A partir de la edad de 34,7 años, los tiempos aumentan progresivamente. También el número de participantes disminuye con la edad.

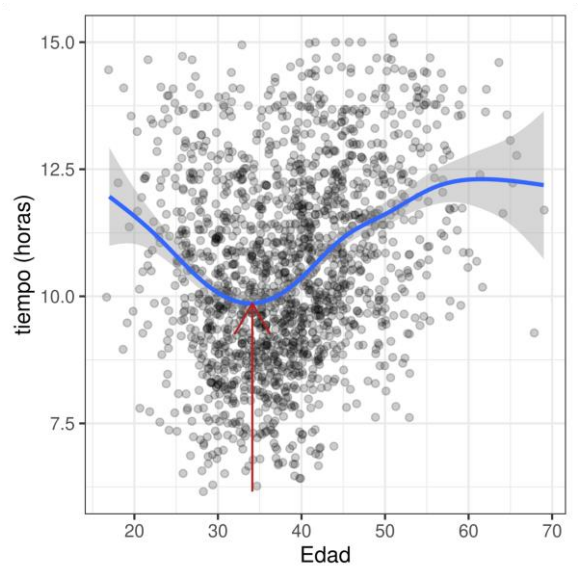

Figura 10: tiempo (horas) en completar la carrera, por edad. Fuente: elaboración propia.

Cumplimiento de la normativa del Reglamento General de Protección de Datos

El conjunto de datos utilizado en este TFM cumple con la normativa del Reglamento General de Protección de Datos. Los datos personales utilizados en este TFM han sido anonimizados previo a su uso, quedando excluidos de la aplicación de la normativa del Reglamento General de Protección de Datos.

4.3. Construcción y evaluación de los modelos predictores

Se han utilizado *Random Forests* y *Redes Neuronales* en una conformación de *Perceptrón Multicapa* para investigar cuál de estos dos modelos generaliza mejor los tiempos de la carrera *Pels Camins dels Matxos*. El proceso de construcción y evaluación del modelo se resume en la Figura 11. Para ello, se ha dividido el conjunto de datos de entrada transformados en un conjunto de entrenamiento con el 75% de las instancias, reservando el 25% restante para la evaluación. Seguidamente se construyeron:

i) un modelo predictor utilizando *Random Forests*, se evaluó el modelo, y se intentó mejorar la selección de *hiperparámetros* probando distintas configuraciones de número de árboles (100, 500, 1.000), variables predictoras seleccionadas al azar en cada bifurcación (3, 5), mínimo número de instancias en los nodos terminales (5, 8) y profundidad máxima del árbol;

ii) un modelo predictor utilizando un *Perceptrón Multicapa*, se evaluó el modelo, y se intentó mejorar la selección de *hiperparámetros* probando distintas configuraciones de número de capas ocultas (1, 2), neuronas por capa (32, 64), ratio de entrenamiento, número de épocas, función de error y optimizador de la función de error.

Finalmente, se eligió el mejor modelo para utilizarlo en la carrera real.

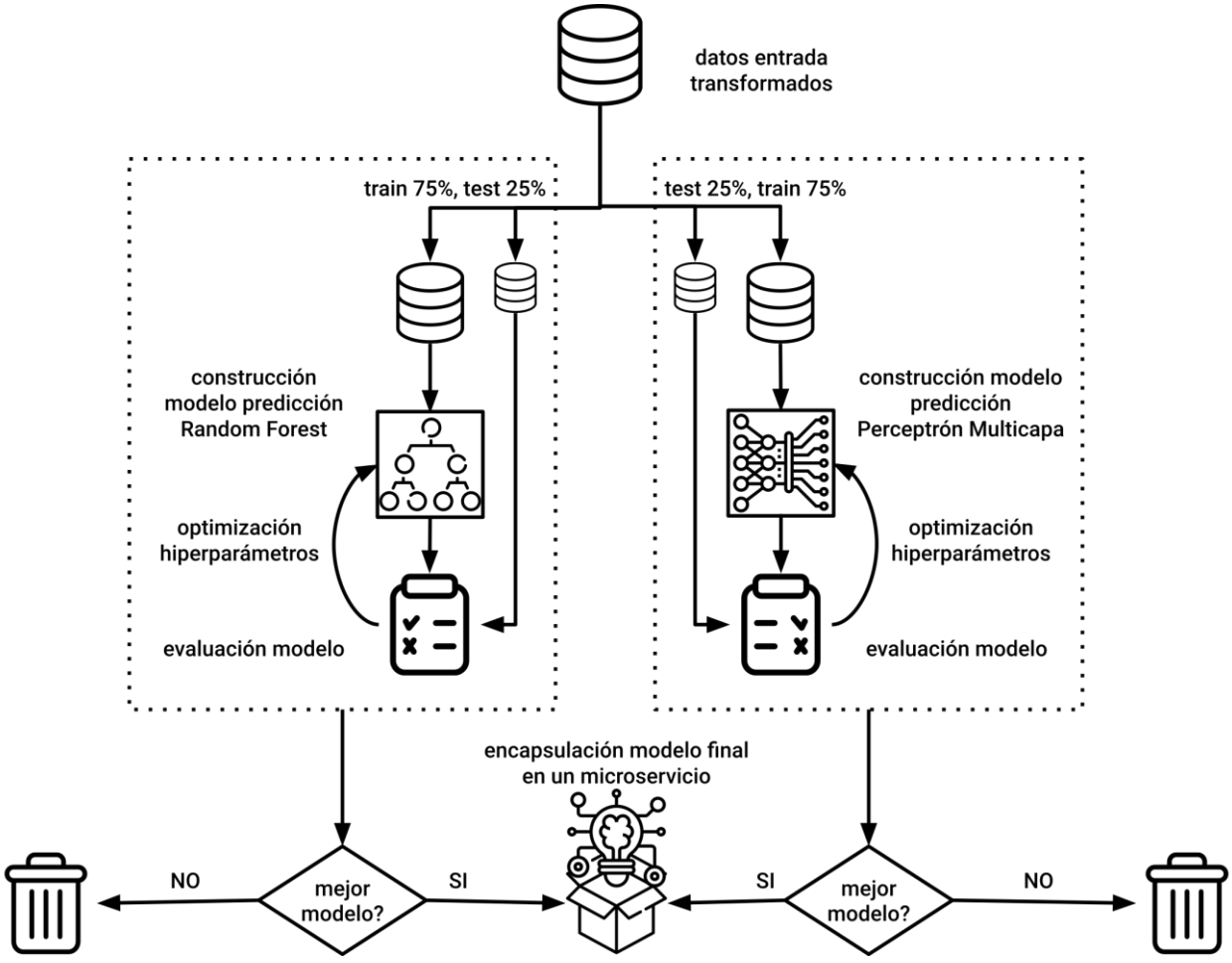

Figura 11: Resumen del proceso de construcción y evaluación del modelo predictor que se utilizará para la carrera. Fuente: elaboración propia.

Nuestro conjunto de datos define una serie temporal, y para ello es necesario crear una arquitectura de predicción que permita generalizar el comportamiento *autoregresivo* de los datos. La predicción de un control depende de la predicción del control anterior, y al mismo tiempo se va a utilizar como entrada para el siguiente control (Figura 12). Esto implica construir un modelo separado para cada control *i* el cual va a predecir el tiempo para el control *i+1*, dando lugar a tantos modelos predictores como controles intermedios hay en la carrera. Durante la carrera, cuando un corredor pasa por el control *j*, utilizando el modelo del control *j* se estima el tiempo del control *j+1* usando todos los tiempos conocidos hasta el control *j*. Sucesivamente se llama al resto de modelos usando todos los tiempos conocidos hasta el control *j* y los tiempos estimados para cada control a partir de *j+1*. Esto define una estrategia de predicción iterativa en la que se utiliza una combinación de tiempos conocidos y tiempos estimados para la estimación del tiempo del siguiente control.

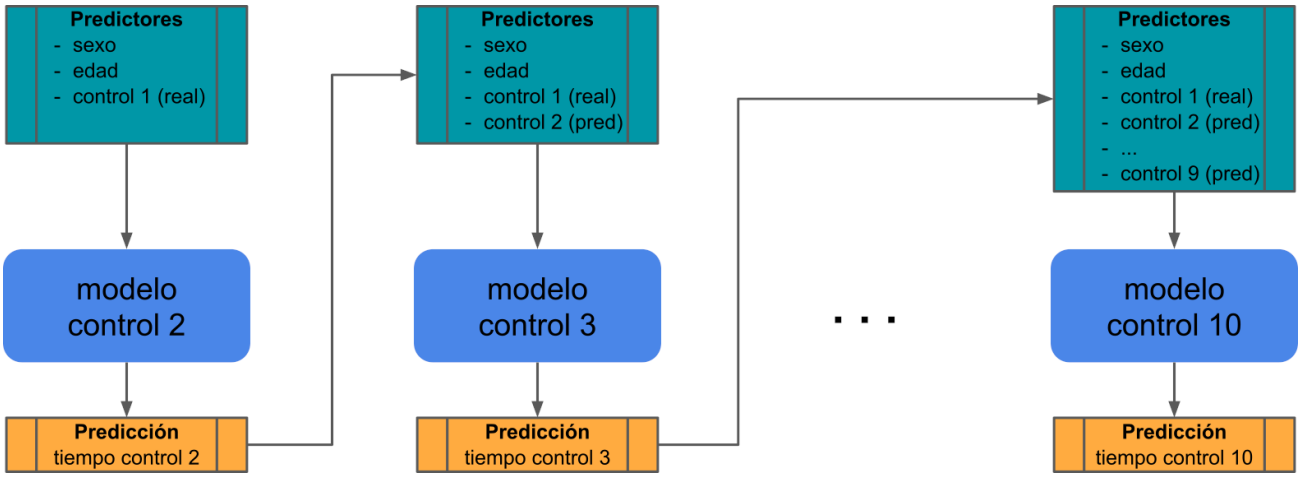

Figura 12: Estrategia de predicción iterativa. Fuente: elaboración propia.

Para la evaluación de los modelos se ha calculado para cada control el *Mean Absolute Error* (MAE, error absoluto medio) de la predicción de las instancias del conjunto de validación. Este se calcula como:

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

Ecuación 1: Error Absoluto Medio. Fuente: https://en.wikipedia.org/wiki/Mean_absolute_error.

Donde *n* son todas instancias del conjunto de validación, y_i el valor real de tiempo de la instancia *i* para un control, y x_i el valor estimado por el modelo. Este expresa el error medio del modelo en unidades de la variable objetivo de la predicción (segundos, en este caso), facilitando su interpretación.

4.3.1. Random Forests

Random Forests es un algoritmo de aprendizaje supervisado que combina la predicción de múltiples árboles de decisión. Durante la fase de entrenamiento el algoritmo "crece" múltiples árboles, cada uno utilizando un subconjunto diferente de los datos de entrenamiento y de variables predictoras (Figura 13A). Durante la fase de predicción la nueva instancia es evaluada independientemente por cada árbol y la predicción final es el resultado de agregar el nodo terminal de cada árbol (Figura 13B). Para problemas de clasificación, la función de agregación suele ser el voto de la mayoría, mientras que para problemas de regresión suele tomarse la media de los valores del nodo terminal de cada árbol. Esta técnica se llama *bootstrap aggregation*, o *bagging*, y pretende minimizar el error en la predicción producido en un único árbol, sin que esto suponga subreajustar la predicción a los datos del conjunto de entrenamiento.

Random Forests tiene ciertas ventajas con respecto a otros métodos de aprendizaje automático (Tabla 3). Las principales características que hacen de *Random Forests* un algoritmo apto para aplicar al problema de la predicción de tiempos planteado en este TFM, son: i) sirve para resolver problemas de regresión; ii) permite identificar relaciones no lineales entre la variable dependiente y las variables predictoras; iii) permite combinar variables predictoras cualitativas (sexo), discretas (edad) y continuas (tiempo de paso en un control) en un mismo modelo de predicción.

Ventajas	Desventajas
• Soluciona los problemas de sobreajuste de los árboles de decisión. • Es apto para problemas de clasificación y regresión. • Permite combinar variables predictoras cualitativas y contínuas. • Funciona bien para conjuntos de datos (casi) de todos los tamaños: desde relativamente pequeños hasta muy grandes. • Resuelve problemas lineales y no lineales. • Es rápido de entrenar.	• Sus resultados son complicados de interpretar. • No funciona bien para conjuntos de datos pequeños.

Tabla 3: Ventajas y desventajas del algoritmo de aprendizaje Random Forests. Fuente: elaboración propia.

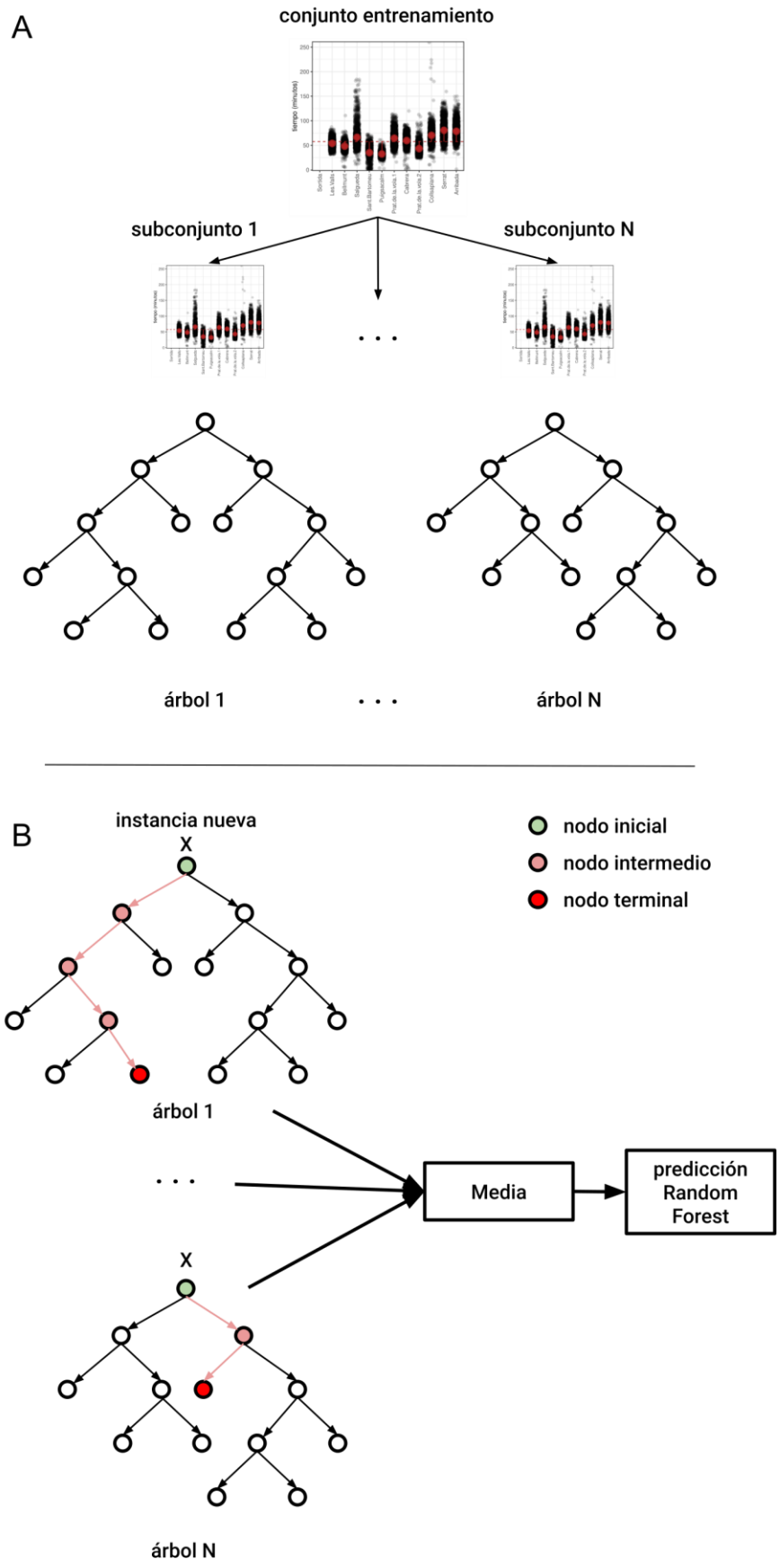

Figura 13: Proceso de entrenamiento y predicción del algoritmo de Random Forests A: proceso de entrenamiento. B: proceso de predicción de un valor continuo para una nueva instancia. Fuente: elaboración propia.

Para entrenar el modelo predictor mediante el algoritmo de *Random Forests*, se utilizó el entorno estadístico R y las funciones implementadas en el paquete *randomForest* (Liaw & Wiener, 2002). La estrategia de creación de modelo predictor está resumida en la Figura 14. Para cada control se utilizó como variables predictoras la edad, el sexo, y el número de segundos transcurridos desde el inicio de la carrera para todos los controles por los que los corredores han pasado. Esto va a generar un modelo predictor para cada control *n* que estima el tiempo del control *n+1*. Nuestro modelo predictor es la colección de los modelos de cada control.

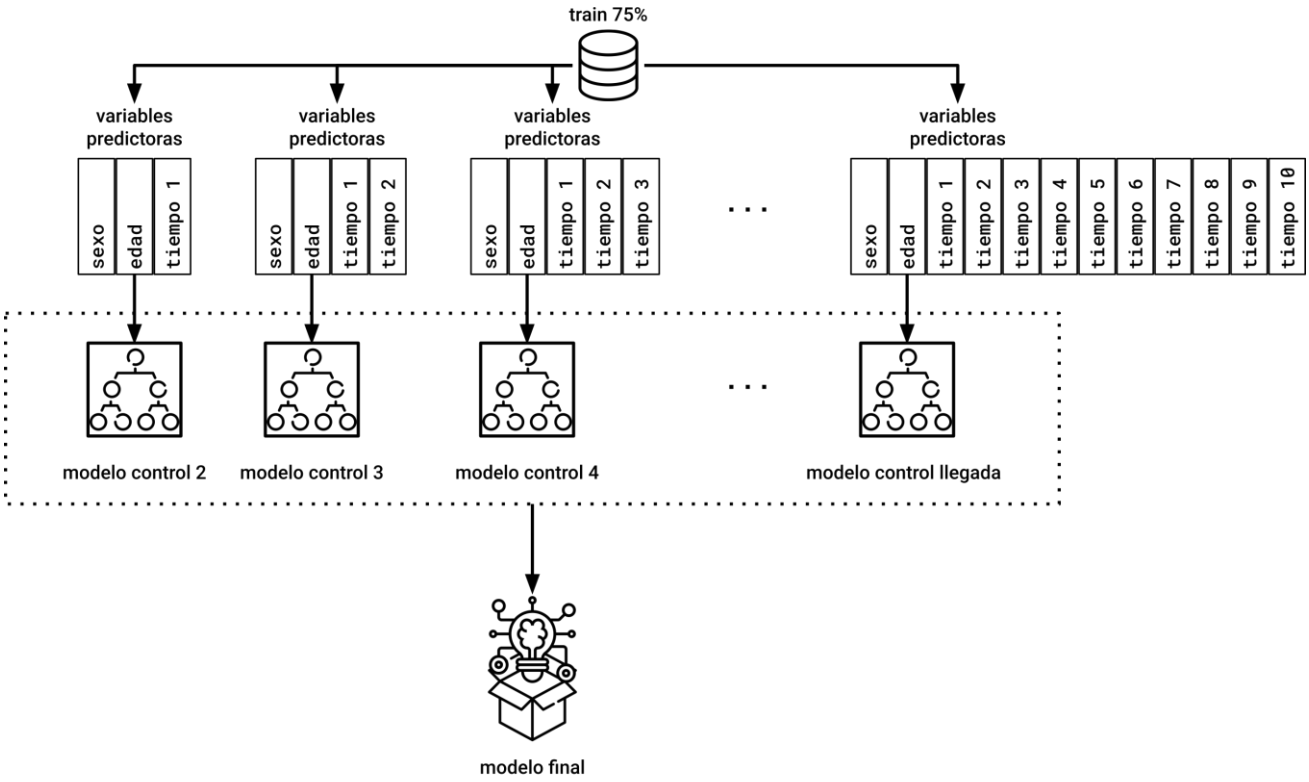

Figura 14: Estrategia de entrenamiento del modelo predictor. Fuente: elaboración propia.

La validación del modelo se realizó calculando el *Mean Absolute Error* (MAE) para cada control estimado desde los controles anteriores, tal y como se describe en la Figura 15. Para un control *n* (*n*=1 en la figura), se predice el tiempo del control *n+1* utilizando el modelo predictor para este control, y el resultado se añade como variable predictora para el siguiente control *n+2*. Este proceso se realiza de forma iterativa hasta llegar a predecir el control de Llegada. Esto nos permite estimar el error que se produce en la predicción de cada control, proporcionando un valor estimado y su error asociado por cada posible control de origen desde donde se realiza la predicción.

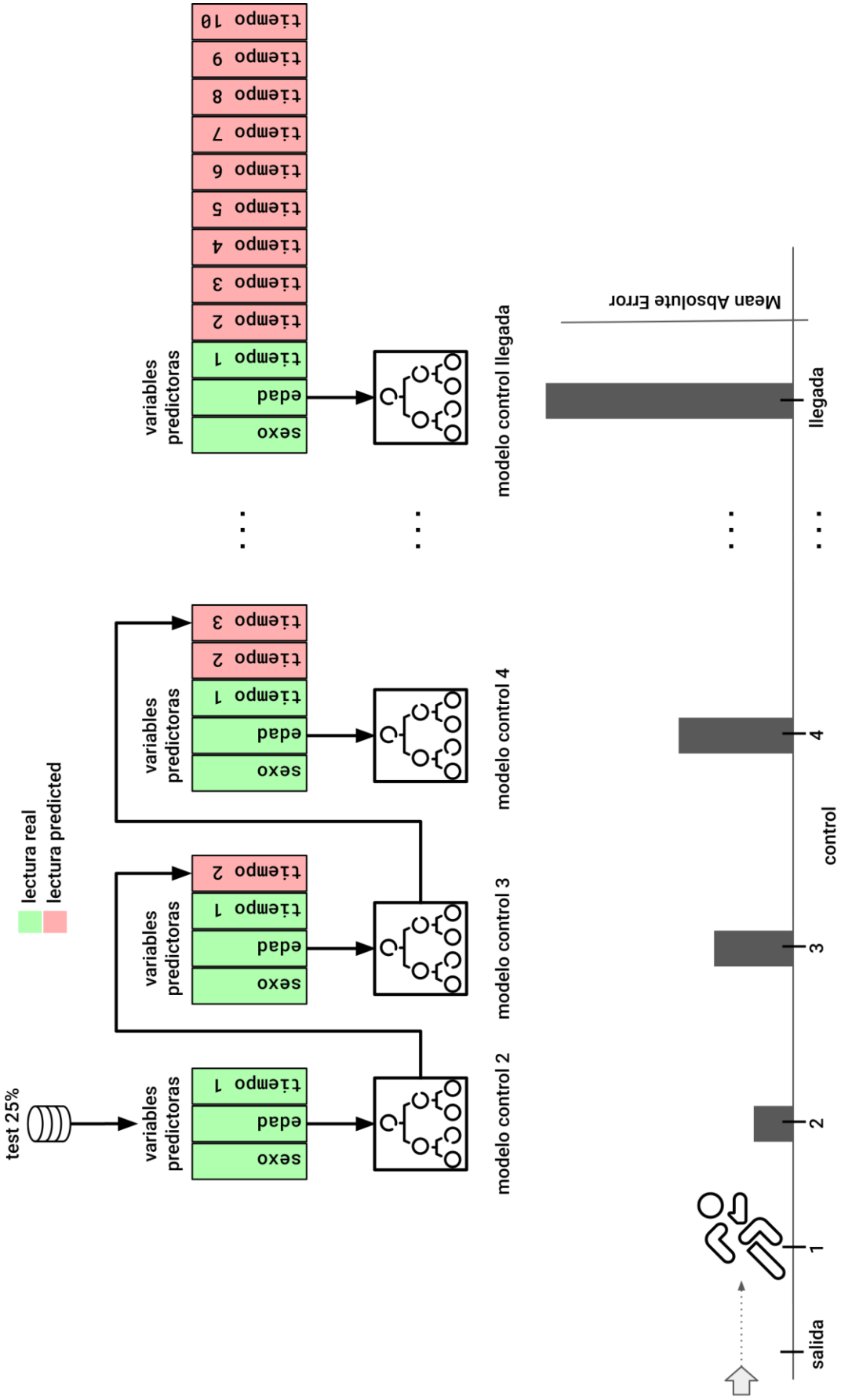

Figura 15: Estrategia de validación del modelo predictor. Fuente: elaboración propia.

Se probaron diferentes configuraciones de *hiperparámetros* para obtener el mejor modelo de regresión. Como métrica de "mejor modelo" se decidió utilizar el *MAE* del control de Llegada utilizando como último control conocido el control intermedio número 5 (Puigsacalm). La estrategia para optimizar los *hiperparámetros* fue utilizando una estrategia de *trial and error*, es decir, simplemente probando distintos valores y guardando los que mejor se comportaban. En la siguiente tabla se resumen los *hiperparámetros* probados y el MAE:

	ntree	mtry	nodesize	maxnodes	MAE
modelo 1	100	3	5	Inf	44,2
modelo 2	100	5	5	Inf	40,7
modelo 3	500	3	5	Inf	22,4
modelo 4	500	5	5	Inf	27,6
modelo 5	1000	5	8	Inf	35,2

Tabla 4: hiperparámetros usados en Random Forests y MAE para la predicción del tiempo de llegada desde el control número 5 (Puigsacalm). Fuente: elaboración propia.

Donde *ntree* son el número de árboles a entrenar; *mtry* son el número de variables predictoras seleccionadas al azar en cada bifurcación; *nodesize* son el número mínimo de instancias en los nodos terminales; *maxnodes* son el máximo número de nodos terminales que puede tener un árbol, que en definitiva es una forma de limitar la profundidad máxima a la que puede crecer un árbol.

El modelo que predice con menor error fue el modelo número 3 (Tabla 4), que produce un error absoluto medio de 22,4 minutos en la estimación del tiempo de llegada desde el control intermedio número 5. Este modelo utiliza como *hiperparámetros* 500 árboles, 3 variables predictoras en cada bifurcación, un mínimo de 5 instancias en el nodo terminal y ninguna restricción en la profundidad delos árboles.

La Figura 17 describe el error absoluto medio en la estimación del tiempo de paso en los controles que restan para finalizar la carrera desde un control origen. El título de cada gráfico describe el control origen desde donde se realiza la predicción. Los gráficos están ordenados por distancia respecto al control de salida, de izquierda a derecha, y de arriba a abajo. Se puede observar que los errores decrecen a medida que avanza el corredor en la carrera, al tiempo que se incorporan más datos reales. Tomando como referencia el control intermedio número 5 (Puigsacalm), el MAE a la Llegada es de únicamente 22,4 minutos. Este error

representa únicamente un 0,37% de desviación respecto a los 6.031 minutos de media que tardan los corredores en completar la carrera.

Una característica interesante del algoritmo de *Random Forests* es que permite calcular la importancia de las variables predictoras dentro del modelo (Breiman, 2001). Esto se debe al proceso de *bagging* descrito anteriormente, en que se crecen árboles utilizando diferentes subconjuntos de instancias y variables predictoras para el entrenamiento. Esto permite comparar la precisión del subconjunto de árboles que incluyen una variable respecto a los que no la incluyen. De esta comparación se puede calcular el incremento del error que implica no incluir una variable concreta. Un mayor error significaría que la variable en cuestión tiene una mayor importancia para conseguir mejor precisión en la estimación.

El la Figura 16 se muestra el porcentaje de incremento del error cuadrado medio en la predicción del tiempo de llegada del corredor cuando una variable predictora no es incorporada en el modelo. Las variables más importantes son los tiempos en los controles previos a la llegada (Collsaplana 22,1%, Serrat 24,9%), y el sexo la menos importante (4,1%). Cabe destacar que el control de salida no tiene ninguna importancia ya que todos los corredores salen en el mismo momento.

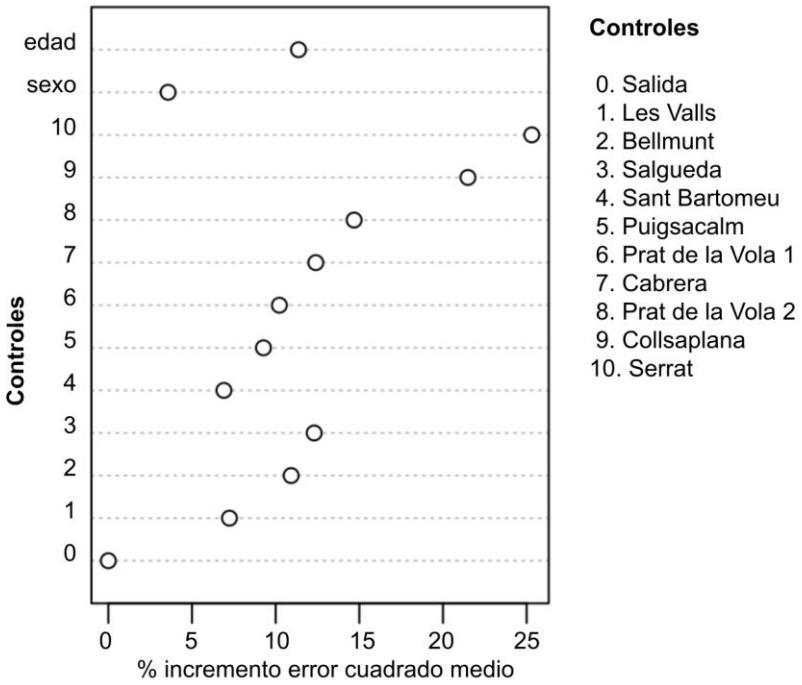

Figura 16: importancia de las variables en el modelo predictivo. Fuente: elaboración propia.

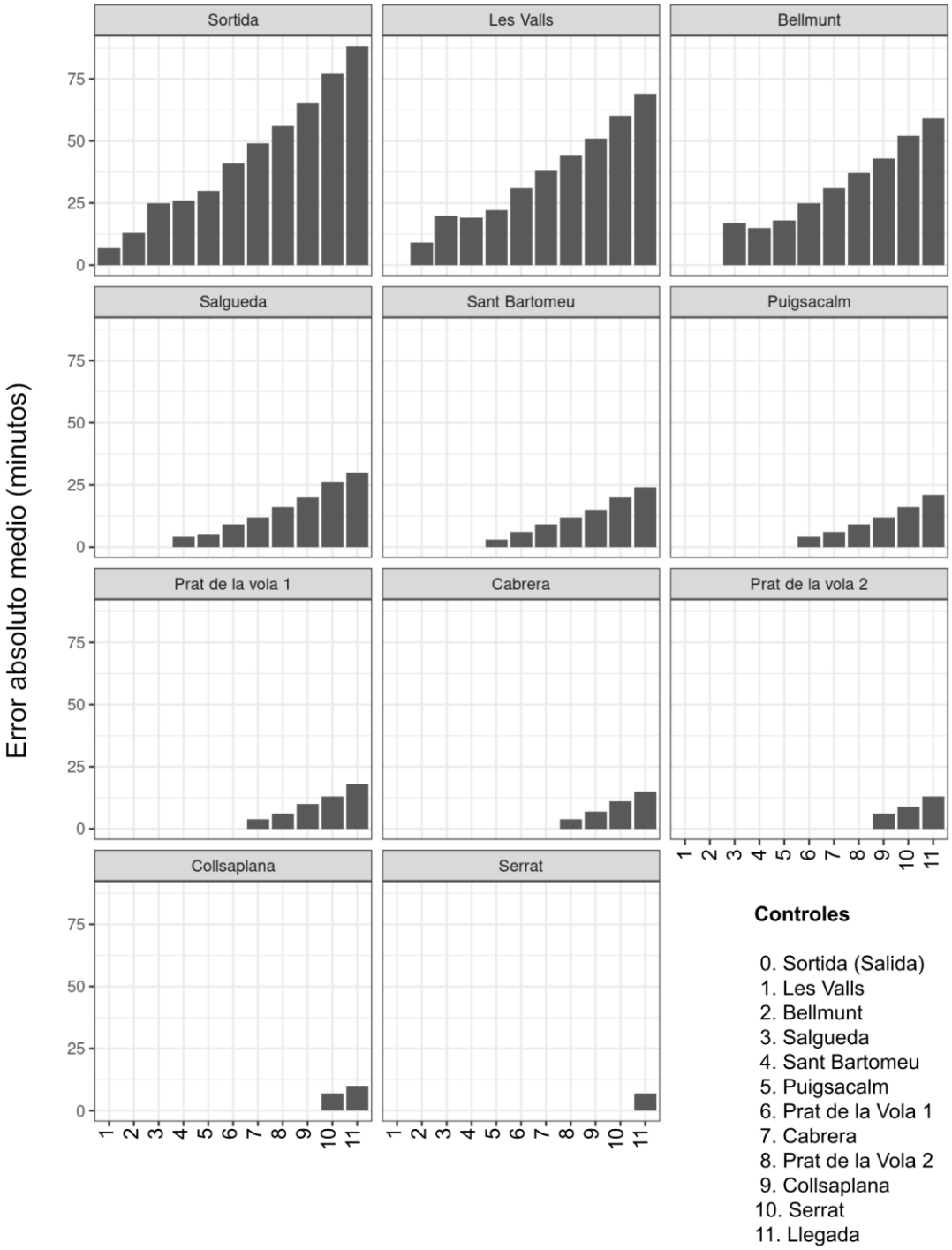

Figura 17: Error medio absoluto en la predicción desde un control a los siguientes controles. Los controles estan ordenados de izquierda a derecha y de arriba a abajo, por orden de kilometrage. Fuente: elaboración propia.

4.3.2. Perceptrón Multicapa

El *Perceptrón Multicapa* es una clase de red neuronal formada por varias capas de nodos (o neuronas): una capa de entrada, una o más capas ocultas, y una capa de salida. Es de tipo *feedforward*, en que las conexiones entre los nodos avanzan siempre en una única dirección sin formar un ciclo. El número de nodos de la capa de entrada coincide con el número de variables predictoras. En la capa de salida, el número de nodos depende de si se trata de un problema de regresión (el valor de salida es recogido por un único nodo) o de clasificación (la probabilidad de cada clase posible es recogida por un nodo distinto). El número de capas ocultas y sus nodos es un *hiperparámetro* del modelo, a optimizar para cada problema que se pretenda resolver.

El proceso de entrenamiento de un Perceptrón Multicapa se basa en ajustar de manera iterativa los pesos de las conexiones de cada nodo con el siguiente, de forma que minimicen el error en la capa de salida (Figura 18C). Esta técnica se llama propagación hacia atrás del error (*backpropagation*), ya que el proceso empieza por la última capa y acaba en la primera. El Perceptrón Multicapa funciona bien para problemas de clasificación y regresión, utilizando una combinación adecuada de funciones de activación y pérdida (Figura 18A, Tabla 5). En problemas de regresión, la función de activación de la neurona será *Linear* si el rango de valores de salida va de [-Inf, +Inf], o *ReLU* si va de [0, +Inf] (Figura 18B). La función de activación simplemente describe la transformación que se aplicará al valor de salida de un nodo, para conseguir una respuesta dentro de un rango concreto de valores.

Tipo de problema	Tipo de salida	Función activación	Función pérdida
Regresión	Valor numérico	Linear o ReLU	Mean Squared Error
Clasificación	Binario (true/false)	Sigmoide	Binary Cross Entropy
Clasificación	Múltiples clases, pertenencia a una única clase	Softmax	Cross Entropy
Clasificación	Múltiples clases, pertenencia a zero o más clases	Sigmoide	Binary Cross Entropy

Tabla 5: Funciones de activación y de pérdida según el tipo de problema. Fuente: https://towardsdatascience.com/deep-learning-which-loss-and-activation-functions-should-i-use-ac02f1c56aa8

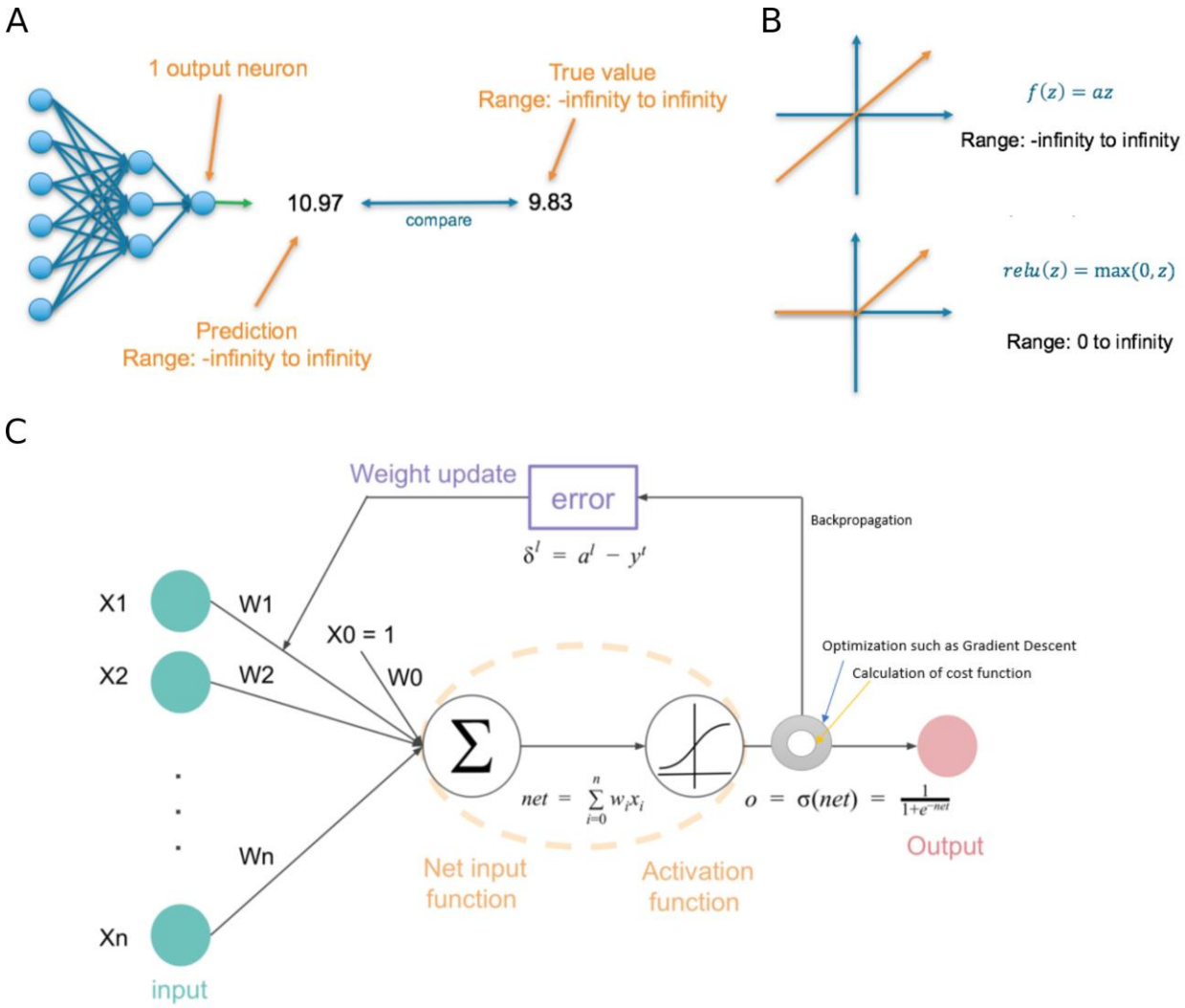

Figura 18: A: Esquema de un perceptrón multicapa para un problema de regresión. B: funciones de activación para un problema de regresión. C: funcionamiento esquemático del método de propagación hacia atrás del error. Fuentes: https://towardsdatascience.com/deep-learning-which-loss-and-activation-functions-should-i-use-ac02f1c56aa8 y https://datascience.stackexchange.com/questions/44703/how-does-gradient-descent-and-backpropagation-work-together

El *Perceptrón Multicapa* tiene características que lo hacen un método que potencialmente puede solucionar el problema que planteamos, ya que permite resolver problemas de regresión en que no exista una relación lineal entre la variable dependiente y las variables predictoras. Además, permite combinar variables predictoras de tipo continuo y discreto en un mismo modelo. Las principales desventajas son la difícil interpretación de las estimaciones que propone, y que es necesario un volumen relativamente importante de datos de entrenamiento. Sus características quedan resumidas en la Tabla 6:

Ventajas	Desventajas
• Es apto para problemas de clasificación y regresión. • Funciona bien para problemas muy grandes. • Resuelve problemas lineales y no lineales. • Puede resolver problemas muy complicados.	• Sus resultados son muy complicados de interpretar. • No funciona bien para conjuntos de datos pequeños. • Es computacionalmente costoso de entrenar.

Tabla 6: Ventajas y desventajas del algoritmo de aprendizaje del Perceptrón Multicapa. Fuente: elaboración propia.

Para entrenar el modelo predictor mediante un *Perceptrón Multicapa*, se utilizó el *software* R y las funciones implementadas en los paquetes *Keras* (Allaire & Chollet, 2021) y *Tensorflow* (Abadi et al., 2016). Como conjuntos de entrenamiento y test se utilizaron los mismos que para entrenar el modelo con *Random Forests*. También como variables predictoras. Dado que el número de instancias es relativamente bajo para entrenar redes neuronales, las arquitecturas de la red entrenadas fueron de una capa de entrada, una o dos capas ocultas con 32 o 64 neuronas totalmente conectadas con la anterior y siguiente capas, y una capa de salida con una única neurona que contendrá el valor de tiempo estimado para un control (Figura 19). Puesto que vamos a predecir un valor continuo no-negativo, se utilizó la función de activación *ReLU* que transforma a 0 los valores negativos de la capa de salida, mientras que mantiene inalterados los valores positivos.

La estrategia de creación de modelo predictor es exactamente igual que para el modelo de *Random Forests* (Figura 14). Para cada control, se utilizó como variables predictoras la edad, el sexo, y el número de segundos transcurridos desde el inicio de la carrera para todos los controles por los que los corredores han pasado. De esta manera el modelo predictor es una colección de los modelos de cada control. La estrategia de validación del modelo también sigue el mismo método descrito para *Random Forests* (Figura 15).

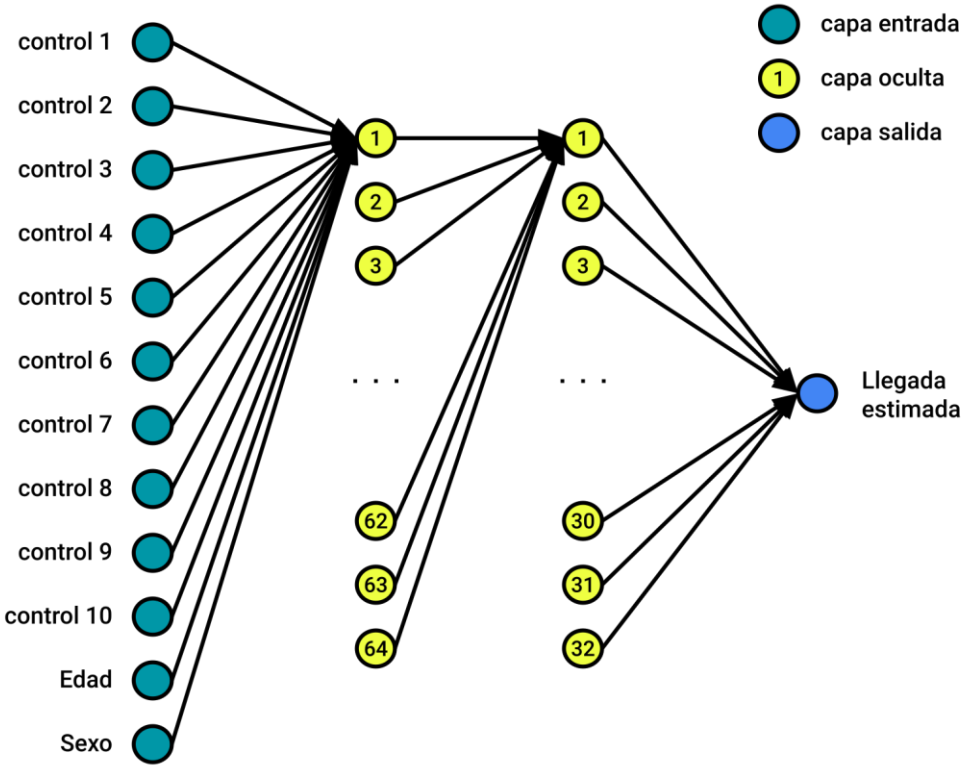

Figura 19: Arquitectura de la red neuronal. Para simplificar la imagen, solo la primera neurona de las capas ocultas está totalmente conectada.

Se probaron diferentes configuraciones de *hiperparámetros* para obtener el mejor modelo de regresión. Como métrica de "mejor modelo" se utilizó, igual que para evaluar el modelo de *Random Forests*, el *MAE* del control de Llegada utilizando como último tiempo conocido el control intermedio número 5 (Puigsacalm). La estrategia para optimizar los *hiperparámetros* fue utilizando una estrategia de *trial and error*, es decir, simplemente probando distintos valores y guardando los que mejor se comportaban. En la Tabla 7 se resumen los *hiperparámetros* probados y el MAE a la llegada desde el control de referencia:

	neuronas		epochs	learning rate	función error	optimizador función error	MAE
	capa 1	capa 2					
modelo 1	32	-	100	0,001	Mean Squared Error	Gradient Descent	29,8
modelo 2	64	-	100	0,001	Mean Squared Error	Gradient Descent	21,2
modelo 3	64	32	100	0,001	Mean Squared Error	Gradient Descent	9,8

Tabla 7: hiperparámetros usados en Perceptrón Multicapa y MAE para la predicción del tiempo de llegada desde el control número 5 (Puigsacalm). Fuente: elaboración propia.

De las configuraciones que se probaron, el modelo que predice con menor error fue el modelo número 3 (Tabla 7), que produce un error absoluto medio de 9,8 minutos en la estimación del tiempo de llegada desde el control intermedio número 5. Este modelo utiliza como *hiperparámetros* 2 capas ocultas con 64 y 32 neuronas respectivamente, 100 *epochs* y un *learning rate* de 0,001, el error cuadrado medio como función de error y *Gradient Descent* para optimizar la función de error.

La Figura 20 describe el error absoluto medio en la estimación del tiempo de paso en los controles que restan para finalizar la carrera desde un control origen. El título de cada gráfico describe el control origen desde donde se realiza la predicción, y en el eje horizontal está el nombre del control para el que se realiza la estimación. Los gráficos están ordenados por distancia respecto al control de salida, de izquierda a derecha, y de arriba a abajo. Se puede observar que los errores no siguen ninguna tendencia, independientemente de la cantidad de datos reales que se utilicen para realizar la estimación. No existe ninguna relación entre el error y la distancia entre el último control del que se dispone el tiempo real y el control sobre el cual se realiza la estimación.

Tomando como referencia el control intermedio número 5 (Puigsacalm), el cual se encuentra aproximadamente a mitad de la carrera, el MAE a la Llegada es de únicamente 9,8 minutos. Este error representa únicamente un 0,16% de desviación respecto a los 6.031 minutos de media que tardan los corredores en completar la carrera.

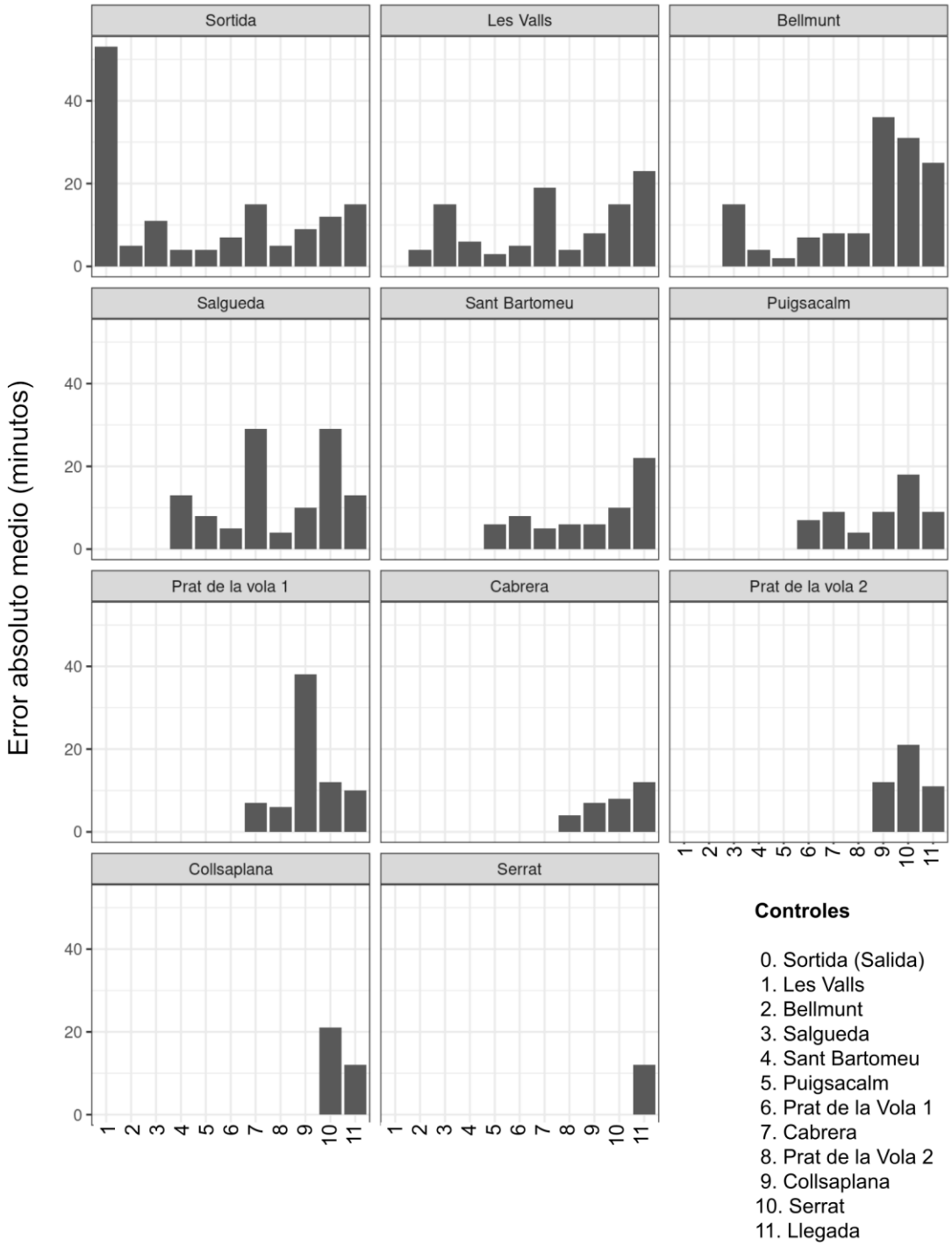

Figura 20: Error medio absoluto en la predicción desde un control a los siguientes controles. Los controles están ordenados de izquierda a derecha y de arriba a abajo, por orden de kilometraje. Fuente: elaboración propia.

4.3.3. Comparación de resultados

Como se muestra en la Figura 21 los dos algoritmos evaluados se comportaron de manera notablemente distinta. Esta figura muestra la evolución del error en la predicción del tiempo de llegada estimado desde los distintos controles intermedios de la carrera. El algoritmo de *Random Forests*, representado por la línea naranja, muestra un error medio absoluto máximo de 87,8 minutos cuando el tiempo de llegada es estimado desde el control de salida (Tabla 8). Esto significa que el modelo realiza la predicción únicamente a partir del sexo y la edad ya que todos los corredores parten en el mismo momento. El error mínimo es de 7,1 minutos, estimando el tiempo desde el último control antes de llegar a la línea de meta (Serrat). La tendencia del error es a la baja y disminuye a medida que se conoce el tiempo real de más controles a lo largo de la carrera. Por otro lado, el algoritmo del *Perceptrón Multicapa*, representado por la línea azul en la figura, muestra un error medio absoluto máximo de 25 minutos estimando el tiempo de llegada desde el control de Bellmunt, en el kilómetro 11 de la carrera (Tabla 8). El error mínimo es de 9,8 minutos estimando el tiempo de llegada desde el control de Puigsacalm en el kilómetro 25. La tendencia del error en la estimación es más o menos plana a partir de este kilómetro, sin llegar a reducirse, aunque se conozcan más tiempos de paso reales por los controles intermedios.

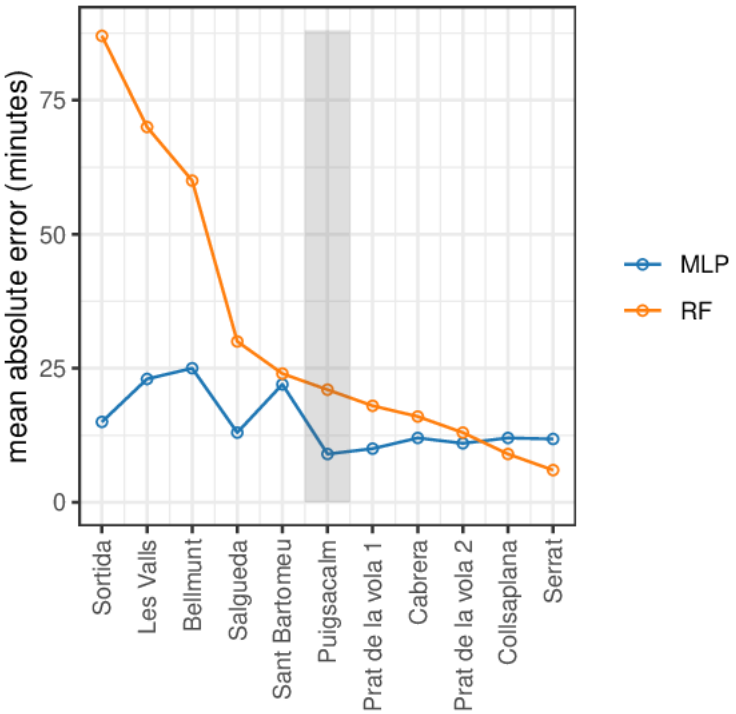

Figura 21: Tiempo de llegada estimado desde los distintos controles intermedios. Fuente: elaboración propia.

Los valores que toma el MAE en el control resaltado en la Figura 21 (Puigsacalm) son los que se utilizaron para determinar la mejor configuración de *hiperparámetros*. Éstos, están resumidos para cada algoritmo en la Tabla 8. La justificación para tomar este control como referencia es que se encuentra en el kilómetro 25 próximo a la mitad de la carrera, en este punto los corredores llevan aproximadamente la mitad del desnivel positivo acumulado, y disponemos de tiempos reales para la mitad de los controles.

	Random Forests	**Perceptrón Multicapa**
MAE llegada mínimo (desde control)	7,1 (Serrat)	9,8 (Puigsacalm)
MAE llegada máximo (desde control)	87,8 (Salida)	25,0 (Bellmunt)
MAE llegada desde Puigsacalm	22,4	9,8
Desviación respecto tiempo medio para finalizar la carrera	0,37%	0,16%
Mejor configuración de *hiperparámetros*	ntree = 500 mtry = 3 nodesize = 5 maxnodes = infinito	neuronas capa 1 = 64 neuronas capa 2 = 32 epochs = 100 learning rate = 0.001 función error = MSE optimizador = Gradient descent

Tabla 8: Resumen de errores y mejores hiperparámetros. Fuente: elaboración propia.

4.4. Encapsulación del modelo predictor en un microservicio

Una vez evaluados los modelos construidos por *Random Forests* y *Perceptrón Multicapa*, se encapsuló el mejor de los modelos en un microservicio que fuera accesible por el protocolo de comunicación HTTP. Este popular protocolo de Internet es muy flexible y se utiliza desde la navegación de páginas web hasta la construcción de web APIs.

El microservicio recibe peticiones para predecir controles por los que un corredor no ha pasado. La aplicación del Centro de Control de Carrera envía una petición cada vez que un corredor pasa por un control. Como entrada, el microservicio recibe todos los tiempos reales por los que el corredor ya ha pasado. Como salida, devuelve todos los controles futuros por los que el corredor no ha pasado.

La estrategia para realizar la predicción es parecida a la utilizada para la validación de los modelos. A partir del *array* de tiempos de un corredor, se predicen los tiempos de manera iterativa desde el primer control desconocido, utilizando como variables predictoras los tiempos de todos los controles anteriores (Figura 22). Como salida, devuelve un *array* con los tiempos reales conocidos y los tiempos estimados por los que no se tenía una lectura real. La aplicación del Centro de Control de Carrera actualiza la base de datos SQL.

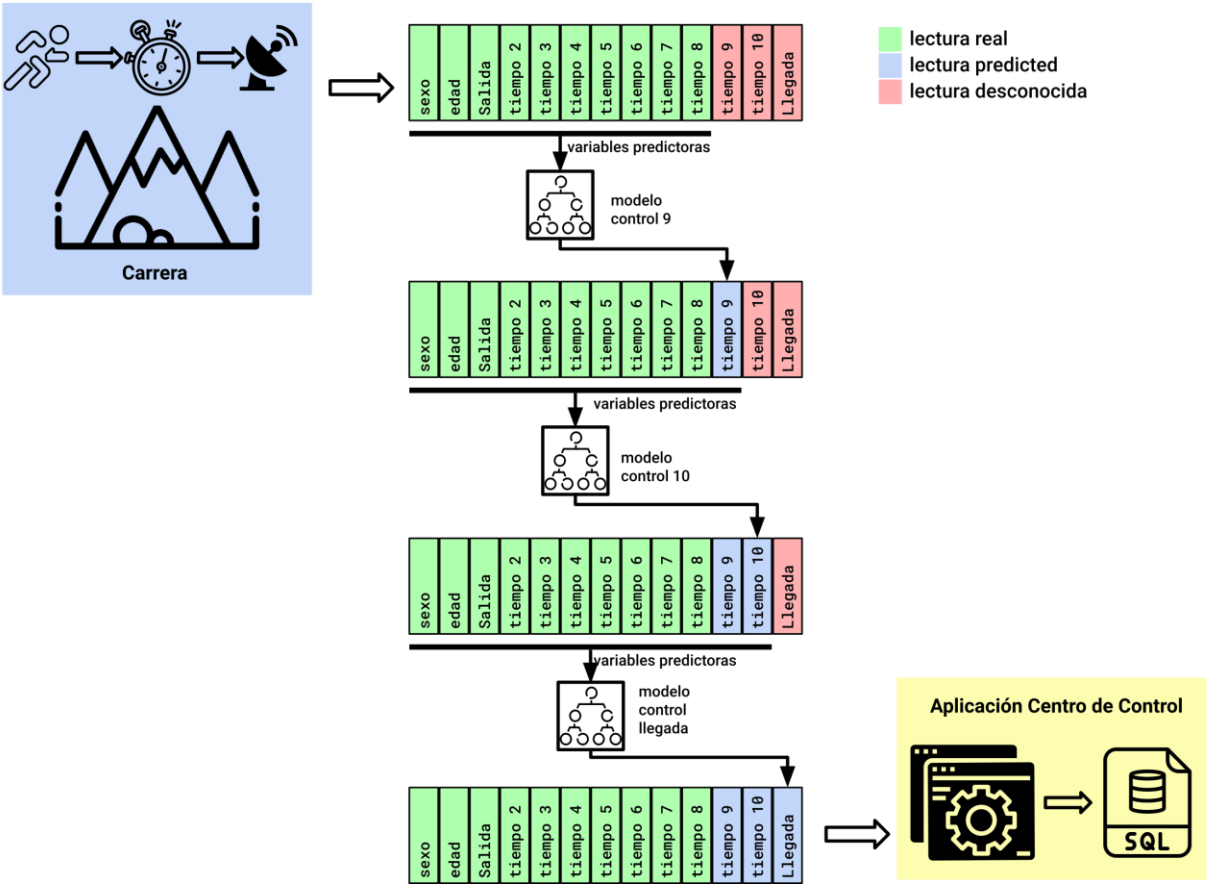

Figura 22: Proceso iterativo de predicción. Después de pasar un corredor por el control 8, el microservicio de predicción recibe una petición para predecir los 3 controles restantes. . Fuente: elaboración propia.

Sin embargo, la aplicación del Centro de Control de Carrera no hace las peticiones directamente a la API de predicciones, sino que simplemente manda un mensaje asíncrono por un *bus* de datos (Figura 23). Se ha implementado un pequeño cliente que “escucha” de este *bus* las peticiones del Centro de Control de Carrera, y las convierte en llamadas a la API de predicción. Una vez recibe la predicción, el cliente envía de retorno por el *bus* el *array* completo con los tiempos reales y estimados para todos los controles de un corredor.

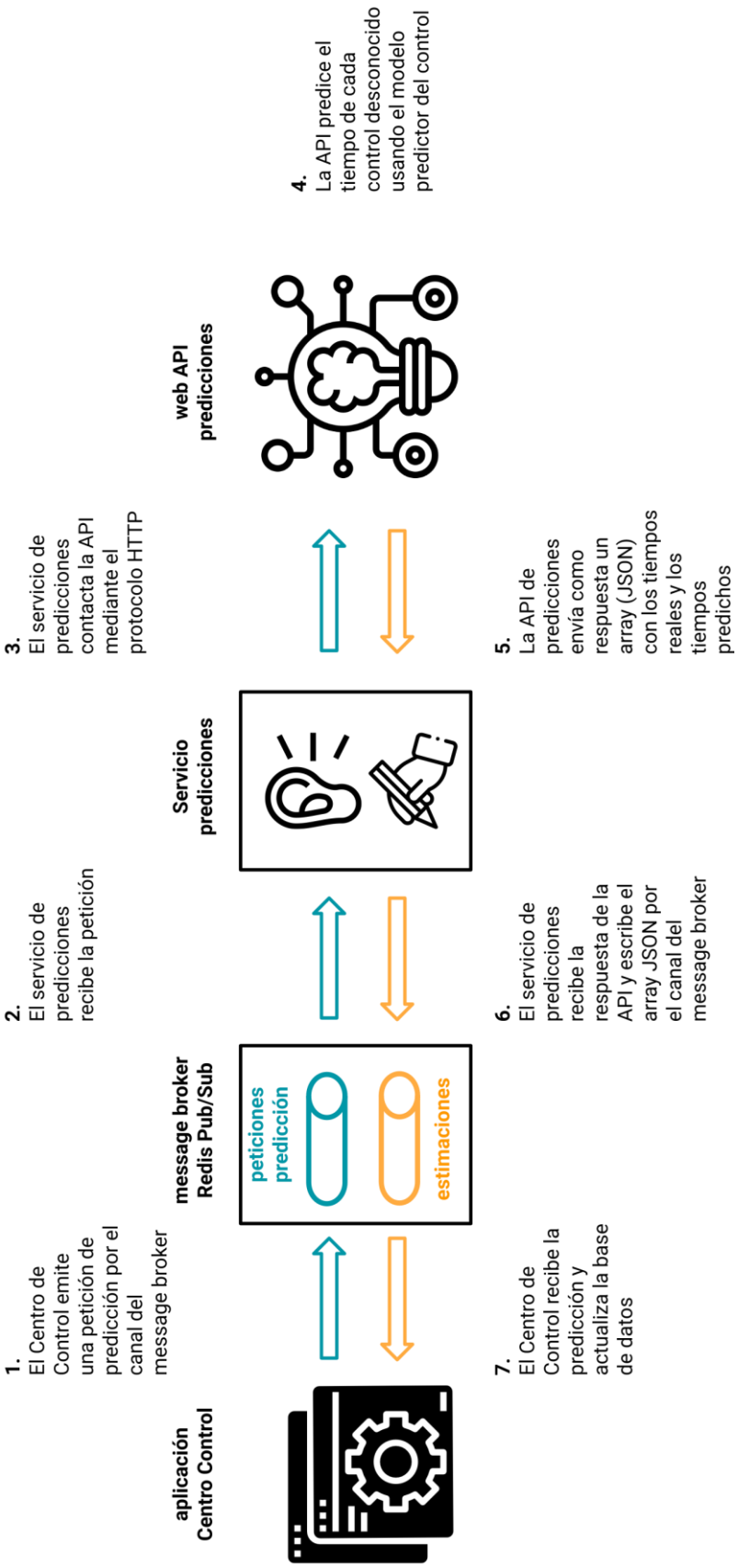

Figura 23: Esquema del protocolo de comunicación entre la aplicación del CCC y la API de predicción. Fuente: elaboración propia.

4.4.1. Construcción de la API web

Una API es el conjunto de definiciones y protocolos que permiten a los sistemas intercambiar información entre ellos. Funciona como mediador entre los clientes y los recursos a los que éstos quieren acceder. Cuando estas API están implementadas sobre el protocolo de Internet HTTP, se dice que son API web.

Para construir la API web se utilizó el entorno R con las funciones implementadas en el paquete *Plumber* (Schloerke, 2020). *Plumber* permite crear de manera muy simple una *API web* decorando el código con comentarios que describen los puntos de entrada y los parámetros de la API. Dado que el modelo predictor está almacenado en un objeto R, y la predicción se realiza llamando a las funciones implementadas en el paquete *randomForest* para R, tener la API también implementada en R facilitó la integración con el resto del conjunto.

La API implementa la estrategia de predicción (Tabla 9). A partir del *array* de tiempos de un corredor, se predicen los tiempos de manera iterativa desde el primer control desconocido, utilizando como variables predictoras los tiempos de todos los controles anteriores. Como salida, devuelve el *array* en formato JSON con los tiempos reales conocidos y los tiempos estimados.

```
peticion = escucha_peticion_http(puerto)

FOR i DE 1 A longitud(peticion.controles) {
    SI VACIO(peticion.controles[i]) {
      peticion.controles[i] = predecir(modelo[i], peticion.controles[1:i-1])
    }
  }

escribe_resultado_http(peticion, puerto)
```

Tabla 9: Pseudocódigo de la API web. Fuente: elaboración propia.

El código del decorador de la función define un método GET para acceder a datos proporcionados por la API mediante el protocolo de comunicación HTTP. Las variables de entrada y de salida son todos los posibles controles de tiempo de la carrera (Tabla 10).

Variable	Descripción	Tipo
SEXO	Sexo del corredor. Valores: (M, H)	Entrada
EDAT	Edad del corredor. Valores: [18, 99]	Entrada
DORSAL	Número de dorsal del corredor. Valores: entero	Entrada/Salida
SOR	Tiempo en segundos desde el inicio de la carrera para el control de Salida. Valores: [0]	Entrada/Salida
VAL	Tiempo en segundos desde el inicio de la carrera para el control de Les Valls. Valores: entero	Entrada/Salida
BELL	Tiempo en segundos desde el inicio de la carrera para el control de Bellmunt. Valores: entero	Entrada/Salida
SAL	Tiempo en segundos desde el inicio de la carrera para el control de Salgueda. Valores: entero	Entrada/Salida
BAR	Tiempo en segundos desde el inicio de la carrera para el control de Sant Bartomeu. Valores: entero	Entrada/Salida
PUI	Tiempo en segundos desde el inicio de la carrera para el control de Puigsacalm. Valores: entero	Entrada/Salida
PR1	Tiempo en segundos desde el inicio de la carrera para el control de Prat de la Vola 1. Valores: entero	Entrada/Salida
CAB	Tiempo en segundos desde el inicio de la carrera para el control de Cabrera. Valores: entero	Entrada/Salida
PR2	Tiempo en segundos desde el inicio de la carrera para el control de Prat de la Vola 2. Valores: entero	Entrada/Salida
COL	Tiempo en segundos desde el inicio de la carrera para el control de Collsaplana. Valores: entero	Entrada/Salida
SER	Tiempo en segundos desde el inicio de la carrera para el control de El Serrat. Valores: entero	Entrada/Salida
ARR	Tiempo en segundos desde el inicio de la carrera para el control de Llegada. Valores: entero	Salida

Tabla 10: Variables de entrada/salida del microservicio de predicción. Fuente: elaboración propia.

La API web espera pues recibir las variables de entrada codificadas dentro de la URL. Por ejemplo, una *query* típica para un control (1) junto con la respuesta del predictor (2), sería:

1. Petición de predicción desde el control 4 (Salgueda) para el dorsal número 6, hombre de 25 años, con el tiempo conocido de los controles de salida (SOR=1), les Valls (VAL=3000) y Salgueda (SAL=9000). Nótese que el control anterior número 3 (Bellmunt) no se conoce, por lo que también será estimado:

```
$ curl http://localhost:3838/pred?EDAD=25&SEXO=M&DORSAL=6&SOR=1&VAL=3000&SAL=9000
```

2. El *string* JSON de salida retornada por el microservicio de predicción. Los tiempos reales y estimados están resaltados en amarillo y azul, respectivamente:

```
{
  "EDAD": 25,
  "SEXO": "M",
  "DORSAL": 6,
  "TIEMPOS": [
    "SOR": 1,
    "VAL": 3000,
    "BELL": 5888,
    "SAL": 9000,
    "BAR": 11569,
    "PUI": 13418,
    "PR1": 17034,
    "CAB": 20562,
    "PR2": 22980,
    "COL": 27010,
    "SER": 31577,
    "ARR": 35718
  ]
}
```

4.4.2. Implementación de un *bus* de mensajes

El *bus* de mensajes se ha implementado usando la herramienta *Redis* (Carlson, 2013). *Redis* es una herramienta que permite guardar en memoria diferentes tipos de estructuras de datos (como listas, mapas, conjuntos, etc.) y sus usos típicos son los de *cache* de datos y *bróker* de mensajes entre aplicaciones. Se utilizó solo un pequeño subconjunto de las funcionalidades de Redis, que serían únicamente sus funciones como *bróker* de mensajes. Un *bróker* de mensajes es simplemente un *software* que funciona como agente de recepción y distribución de mensajes entre sus clientes.

Se creó un canal llamado *predictions_in* en el *bus* de mensajes donde los microservicios de la aplicación del Centro de Control de Carrera pueden enviar las solicitudes de predicción. Las solicitudes se crean insertando un *stream* de texto en formato JSON con los valores de las

variables con las que se debe realizar la predicción. Este canal funciona como entrada al servicio de predicción.

Se creó un canal llamado *predictions_out* donde los microservicios de la aplicación del Centro de Control de Carrera van a recibir las predicciones. En este canal el servicio de predicción escribe un *stream* de texto en formato JSON con los datos de la *query* original completados con los valores estimados por el modelo de predicción. Este canal funciona como salida del servicio de predicción.

4.4.3. Servicio de predicciones

El servicio de predicciones es simplemente el cliente que escucha por el *bus* de mensajes, llama la API de predicción y escribe las predicciones de vuelta al *bus* de mensajes (Tabla 11). Está implementado en R y utiliza las funciones implementadas en el paquete *Redux* (FitzJohn, 2018) para conectarse al *message bróker* como cliente.

El proceso que controla el servicio de predicciones consta de 4 partes:

1. *Listener*: conecta al canal *predictions_in* a la espera de la llegada de alguna petición de predicción.
2. *Input parser*: valida y convierte el *stream* de entrada de formato JSON a una lista de variables predictoras.
3. Predicción: llamada a la *API web* de predicción con las variables predictoras. Para ello se utilizará el método GET del protocolo HTTP para acceder a la API.
4. Writer: conversión de las variables predichas a formato JSON y escritura al canal de salida *predictions_out*.

Está implementado según el siguiente fragmento de pseudocódigo:

```
peticion = subscribe_Redis(“predictions_in”)  // subscripción al canal de entrada Redis

SI es_valida(peticion) {                      // valida formato peticion enviada por el CCC
  pred = hacer_peticion_http(peticion, api_pred, “GET”) // envia la peticion a la API web
}

publish_Redis(toJSON(pred),”predictions_out”) // escribe las predicciones al canal de salida
```

Tabla 11: Pseudocódigo del servicio de predicciones. Fuente: elaboración propia.

4.5. Discusión

A partir del análisis de los datos de las ediciones anteriores de la carrera, se han confirmado algunas de las observaciones afirmadas por la literatura científica revisada. Los datos de la carrera *Pels Camins dels Matxos* sugieren que mujeres y hombres necesitan un tiempo distinto para completar la carrera. En concreto, las mujeres necesitan 1 hora y 17 minutos más que los hombres en completar el recorrido, y muestran una mayor dispersión respecto al tiempo medio para completar un control. En lo que respecta a la edad, la probabilidad de finalizar con el menor tiempo es más alta alrededor de los 34,7 años. Estas observaciones coinciden con las conclusiones de los trabajos realizados por Knechtle et al. (2014) y Alvero-Cruz et al. (2019) donde utilizaban el sexo y la edad como variables predictoras en modelos de regresión lineal múltiple para estimar los tiempos de llegada de los corredores.

Desafortunadamente, nada podemos decir del resto de variables antropométricas utilizadas en estos estudios (índice de masa corporal (BMI), capacidad pulmonar (VO_2 max), pulsaciones por minuto máximas y nivel de lactato en sangre). Estas son difíciles de conseguir sistemáticamente, especialmente en corredores amateurs de carreras populares. Igualmente, los datos meteorológicos de las ediciones anteriores no están disponibles, y sería especialmente complicado registrarlos regularmente (cada hora, por ejemplo) para cada control de la carrera.

En este trabajo se ha demostrado que la predicción de los tiempos de paso a partir del sexo, la edad y los tiempos de paso intermedios es posible. Los errores en las estimaciones de los algoritmos probados son aceptablemente bajas (0,16% - 0,37% de error respecto al tiempo medio para finalizar la carrera). Sin embargo, ha sido necesario utilizar algoritmos que tengan la característica de permitir utilizar una combinación de variables continuas y categóricas para realizar la predicción. Utilizando el algoritmo de *Random Forests* el error fue de sólo 22 minutos de media respecto los 6.031 minutos de media que tardan los corredores en terminar la carrera. Utilizando el algoritmo del *Perceptrón Multicapa* el error es de sólo 10 minutos de media.

Una característica importante es la tendencia de los errores de predicción a medida que la carrera avanza y se conocen más tiempos reales de los corredores. Según la estrategia seguida para construir los predictores, se espera que el error en la predicción sea mayor cuanto más lejos esté situado el control estimado con respecto al último control con tiempo conocido. Igualmente, de manera intuitiva se espera que el error disminuya a medida que los predictores se vayan nutriendo de más datos reales. No es lo mismo realizar la predicción de la llegada cuando un corredor ha pasado por solo 2 controles intermedios, que cuando ha

pasado ya por 5. Esta tendencia en la evolución del error se ha visto claramente utilizando el algoritmo de *Random Forest*. En este caso, el error medio a la llegada disminuía a medida que se avanzaba el control desde el cual se realizaba la predicción. Sin embargo, utilizando el *Perceptrón Multicapa* el error no seguía ninguna tendencia y era inmune al hecho de predecir un control más cercano, o utilizando más datos reales conocidos.

La inesperada tendencia de los errores que reportó el algoritmo del *Perceptrón Multicapa* fue el principal motivo para no encapsularlo en el servicio de predicción que se utilizará en el entorno de producción.

De todos modos, los bajos errores que obtenemos con este algoritmo sugieren que existe un potencial en utilizar *redes neuronales* para la predicción de los tiempos de paso. Dado el carácter *autoregresivo* de los datos, perecido a una serie temporal, puede que el *Perceptrón Multicapa* no ser la mejor arquitectura de red neuronal para este tipo de problema. Otras arquitecturas como las *Redes Neuronales Recurrentes* (RNN) han demostrado un buen desempeño en la resolución de problemas descritos como series temporales (Dhall et al., 2020). Éstas, utilizan memoria para guardar el estado de la red en un momento anterior en el tiempo, que se utiliza como entrada para relacionar la salida con el estado anterior. De todas maneras, la utilización de una arquitectura de red neuronal distinta no es trivial y requiere replantear la estrategia de entrenamiento del modelo predictor, así como un detallado estudio de los *hiperparámetros*.

Respecto a los *hiperparámetros* utilizados en los modelos, se han probado distintas configuraciones siguiendo un criterio meramente intuitivo y basado en la experiencia. Por ejemplo, si el error del *Perceptrón Multicapa* aumentaba cuándo se utilizaban más neuronas en la capa oculta, en la siguiente iteración la construcción del modelo se hacía con un número más pequeño. Aunque este procedimiento de optimización de *hiperparámetros* basado en prueba y error puede funcionar razonable bien, es lento y no garantiza la obtención del conjunto óptimo. Es necesario sistematizar el procedimiento utilizando técnicas formales de optimización de *hiperparámetros* implementadas en *frameworks* de *Inteligencia Artificial* y *machine learning* (Candel & LeDell, 2020).

El número de instancias de las que se disponía para entrenar el modelo era relativamente pequeño respecto a la dimensionalidad del conjunto (1.721 instancias y 13 variables predictoras). Por este motivo, la mejor configuración de *hiperparámetros* para el algoritmo de *Random Forests* fue utilizando un número relativamente pequeño de árboles (500), 3 variables predictoras seleccionadas al azar en cada bifurcación, 5 instancias como mínimo en los nodos terminales y ninguna restricción en la profundidad de los árboles. La mejor configuración de *hiperparámetros* para el algoritmo del *Perceptrón Multicapa* fue utilizando una configuración

de 2 capas ocultas con 64 y 32 neuronas respectivamente, completamente conectadas, y el algoritmo *Gradient Descent* como método optimizador de la función de error. La poca disponibilidad de instancias para el entrenamiento podría justificar los resultados inestables obtenidos con el *Perceptrón Multicapa*. Como hecho positivo, para la próxima edición se va a disponer de aproximadamente 300 instancias adicionales.

5. Conclusiones y trabajo futuro

5.1. Conclusiones

En el presente trabajo se ha logrado desarrollar un piloto de software para estimar el tiempo de llegada de los corredores en la carrera de montaña de *ultradistancia Pels Camins dels Matxos*. Este piloto puede ayudar a los organizadores a mejorar la seguridad de la carrera y anticiparse a posibles accidentes, identificando corredores que no hayan alcanzado un control intermedio en el tiempo estimado.

Se integraron los datos públicos y privados de las distintas ediciones de la carrera. El preprocesado de los datos con técnicas como la discretización, la normalización y los métodos de imputación, es necesario para aplicar correctamente los modelos de regresión probados en este trabajo de investigación. Para la imputación de datos ausentes se ha utilizado el propio conjunto de datos de entrenamiento, pero estrategias distintas para imputar los valores ausentes del sexo (a partir del sexo de otros corredores con el mismo nombre), la edad (a partir de la distribución de la muestra de edades) y los tiempos intermedios (utilizando la proximidad en un *Random Forests* con otras instancias). Se estimó la influencia de las variables del conjunto de datos en la precisión de la estimación, y se concluye que el sexo y la edad son relevantes y mejoran la estimación.

Se determinó la capacidad de los algoritmos de *Random Forests* y del *Perceptrón Multicapa* para predecir los tiempos de llegada a partir de los datos de ediciones anteriores. Se concluye que el error absoluto medio con la mejor configuración de *hiperparámetros* con *Random Forests* es superior a la del *Perceptrón Multicapa* (MAE RF 22,4 minutos frente MAE MLP 9,8 minutos). Sin embargo, con *Random Forests* el valor absoluto del error progresa a la baja a medida que se van incorporando tiempos reales de los corredores mientras que con el *Perceptrón Multicapa* la tendencia es aleatoria.

Se encapsuló el modelo de predicción entrenado con el algoritmo de *Random Forests* en un microservicio web utilizando R y el paquete *Plumber*, obteniendo como resultado un servidor que implementa una API web accesible con el protocolo de internet HTTP. Se concluye que utilizando simples decoradores de código implementados en *Plumber* es posible convertir un modelo de predicción en una API web para ser utilizado en un entorno de producción.

Se implementó un *bróker* de mensajes para facilitar la comunicación entre los demás microservicios de la aplicación del Centro de Control de Carrera y la API de predicción utilizando *Redis* como herramienta. Las aplicaciones se comunican mediante mensajes

codificados en JSON inyectados en los canales *predictions_in* y *predictions_out d*el *bus* de mensajes.

Por consiguiente, se puede concluir que el objetivo principal de desarrollar un prototipo de *software* que permita predecir los tiempos de paso de los corredores por los diferentes controles de la carrera *Pels Camins dels Matxos*, ha sido alcanzado tras la realización de este TFM.

5.2. Líneas de trabajo futuro

El objetivo del trabajo presentado en este TFM fue desarrollar un modelo predictor del tiempo de los corredores, hecho a medida para la carrera *Pels Camins del Matxos*. Una aplicación directa de este trabajo sería en carreras de montaña organizadas por otros clubes, donde se podría adaptar para funcionar integrado con diferentes aplicaciones de control de carrera. Para ello, sería necesario generalizar la parte de construcción del modelo predictor de manera que acepte cualquier número y tipo de variables predictoras que los organizadores deseen utilizar. De todos modos, la parte de integración con la aplicación de control de carrera es complicada de generalizar y sería la única parte que habría que desarrollar a medida para una nueva carrera.

El trabajo futuro se centraría en la predicción de tiempos, pero podría ser extendido para abordar otros problemas que suceden en carrera:

- Predicción de la posibilidad que un corredor abandone la carrera en cualquier momento.
- Identificación de zonas de peligro potencial en el circuito, donde más corredores potencialmente pueden abandonar.
- Detección de fraude en los tiempos de paso por los controles.
- Predicción del tiempo de finalización del corredor más rápido en la *próxima edición*.

Asimismo, durante el desarrollo del trabajo se han identificado limitaciones en el alcance de los objetivos que serían interesantes de estudiar en el futuro:

- El uso de las *redes neuronales* para entrenar el modelo predictor se ha estudiado de manera limitada en una única arquitectura de *Perceptrón Multicapa*. Sería necesario estudiar otras arquitecturas de *redes neuronales* que puedan ser más adecuadas para la predicción de series temporales.

- Explorar el uso de técnicas formales de optimización de *hiperparámetros*, en lugar de la estrategia de *prueba y error* utilizada.

- Estudiar las transformaciones que se pueden aplicar sobre los datos para mejorar las predicciones que realizan los modelos. Cómo afecta la estimación de los modelos cuando los datos son escalados y/o centrados, o se utilizan *offsets* respecto al control anterior en lugar de valores absolutos, es algo que ha quedado fuera de los objetivos del presente trabajo.

6. Bibliografía

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., & others. (2016). Tensorflow: A system for large-scale machine learning. 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), 265–283.

Alvero-Cruz, J. R., Parent Mathias, V., Garcia Romero, J., Carrillo de Albornoz-Gil, M., Benítez-Porres, J., Ordoñez, F. J., Rosemann, T., Nikolaidis, P. T., & Knechtle, B. (2019). Prediction of Performance in a Short Trail Running Race: The Role of Body Composition. Frontiers in Physiology, 10, 1306. https://doi.org/10.3389/fphys.2019.01306

Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

Carlson, J. (2013). Redis in Action (1st ed.). Manning Publications.

Centre Excursionista Torello. (2021). Pels Camins dels Matxos. Retrieved April 21, 2021, from https://www.matxos.cat/

ChampionChip. (2021). Championchip. http://www.championchip.cat/web/

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794. https://doi.org/10.1145/2939672.2939785

Allaire, J. J., & Chollet, F. (2021). keras: R Interface to "Keras." https://CRAN.R-project.org/package=keras

Cleveland, W. S. (1979). Robust Locally Weighted Regression and Smoothing Scatterplots. Journal of the American Statistical Association, 74(368), 829–836. https://doi.org/10.1080/01621459.1979.10481038

Coast, J. R., Blevins, J. S., & Wilson, B. A. (2004). Do Gender Differences in Running Performance Disappear With Distance? Canadian Journal of Applied Physiology, 29(2), 139–145. https://doi.org/10.1139/h04-010

Cuk, I., Nikolaidis, P. T., & Knechtle, B. (2020). Sex differences in pacing during half-marathon and marathon race. Research in Sports Medicine, 28(1), 111–120. https://doi.org/10.1080/15438627.2019.1593835

Dhall, D., Kaur, R., & Juneja, M. (2020). Machine Learning: A Review of the Algorithms and Its Applications. In P. K. Singh, A. K. Kar, Y. Singh, M. H. Kolekar, & S. Tanwar (Eds.), Proceedings of ICRIC 2019 (pp. 47–63). Springer International Publishing.

Ely, M. R., Cheuvront, S. N., Roberts, W. O., & Montain, S. J. (2007). Impact of weather on marathon-running performance. Medicine and Science in Sports and Exercise, 39(3), 487–493. https://doi.org/10.1249/mss.0b013e31802d3aba

FitzJohn, R. (2018). redux: R Bindings to "hiredis." https://CRAN.R-project.org/package=redux

Fogliato, R., Oliveira, N. L., & Yurko, R. (2020). TRAP: A predictive framework for the Assessment of Performance in Trail Running. Journal of Quantitative Analysis in Sports. https://doi.org/10.1515/jqas-2020-0013

Candel, A., & LeDell, E. (2020). Deep Learning with H2O. http://docs.h2o.ai/h2o/latest-stable/h2o-docs/booklets/DeepLearningBooklet.pdf

Hubble, C., & Zhao, J. (2016). Gender differences in marathon pacing and performance prediction. Journal of Sports Analytics, 2(1), 19–36. https://doi.org/10.3233/JSA-150008

ITRA. (2021). International Trail Running Association. Retrieved April 21, 2021, from https://itra.run/

Keogh, A., Smyth, B., Caulfield, B., Lawlor, A., Berndsen, J., & Doherty, C. (2019). Prediction Equations for Marathon Performance: A Systematic Review. International Journal of Sports Physiology and Performance, 14(9), 1159—1169. https://doi.org/10.1123/ijspp.2019-0360

Knechtle, B., Barandun, U., Knechtle, P., Zingg, M. A., Rosemann, T., & Rüst, C. A. (2014). Prediction of half-marathon race time in recreational female and male runners. SpringerPlus, 3(1), 248. https://doi.org/10.1186/2193-1801-3-248

Knechtle, B., Knechtle, P., Barandun, U., Rosemann, T., & Lepers, R. (2011). Predictor variables for half marathon race time in recreational female runners. Clinics, 66(2), 287–291. https://doi.org/10.1590/S1807-59322011000200018

Liaw, A., & Wiener, M. (2002). Classification and Regression by randomForest. R News, 2(3), 18–22.

LiveTrail. (2021). Innovative digital service to track and manage an endurance race. Retrieved April 21, 2021, from https://www.livetrail.net/

Nikolaidis, P. T., & Knechtle, B. (2017). Effect of age and performance on pacing of marathon runners. Open Access Journal of Sports Medicine, 8, 171–180. https://doi.org/10.2147/OAJSM.S141649

R Core Team. (2021). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. https://www.R-project.org/

Rüst, C. A., Knechtle, B., Knechtle, P., Barandun, U., Lepers, R., & Rosemann, T. (2011). Predictor variables for a half marathon race time in recreational male runners. Open Access Journal of Sports Medicine, 2, 113–119. https://doi.org/10.2147/OAJSM.S23027

Schloerke, B., & Allen, J. (2021). plumber: An API Generator for R. https://CRAN.R-project.org/package=plumber

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. Journal of the Royal Statistical Society. Series B (Methodological), 58(1), 267–288.