Facultade de Informática
UNIVERSIDADE DA CORUÑA

TRABALLO FIN DE GRAO
GRAO EN CIENCIA E ENXEÑARÍA DE DATOS

Visualización y uso de técnicas de Aprendizaje Máquina Supervisado para la predicción de resultados de la Fórmula 1

Estudante: Diego Prieto Fernández

Dirección: Manuel Antonio Presedo Quindimil

A Coruña, xuño de 2023.

Dedicado a mis padres, a Irene, a Alexandra y a Kira

Agradecimientos

En primer lugar, agradecer a mi familia por todo el esfuerzo que han realizado durante estos años, por motivarme durante toda la carrera. También quiero agradecer a mi novia Alexandra por animarme a realizar este proyecto y apoyarme durante todo el proceso.

A Fernando Alonso por hacerme fanático de este deporte y a Nicolás Gutiérrez Vázquez por la ayuda e inspiración.

También me gustaría agradecerle a Manuel Antonio Presedo Quindimil la confianza y el apoyo depositado a la hora de ser el tutor de este proyecto.

Por último, quisiera agradecer a mis compañeros de clase, unos buenos amigos que me llevo de la Universidad. Dei, Chou, Brais y Mateo, gracias por hacer que la carrera fuera más entretenida.

A todos ellos, muchas gracias por todo.

Resumen

La Fórmula 1 es considerada como la competición más prestigiosa y tecnológica en el mundo del automovilismo, donde diferentes equipos y sus pilotos compiten por ganar carreras y conseguir el campeonato. Con el uso de datos históricos de carreras previas y mediante la aplicación de modelos de predicción basados en Aprendizaje Máquina Supervisado se pueden predecir los resultados de las siguientes carreras. Este proyecto tiene como objetivo investigar la predicción de los ganadores de los campeonatos, tanto de pilotos como de constructores, proponiendo diversos algoritmos y siguiendo una metodología de minería de datos.

Se presentan una serie de visualizaciones que ayudan a comprender las diferentes variables del conjunto de datos y la historia reciente del deporte. Mediante un preprocesado de los datos y diversas pruebas con los algoritmos propuestos, se consiguieron unos resultados muy alentadores para la predicción de ambos campeonatos. El modelo que mejores resultados obtuvo fue la Red Neuronal Artificial, seguida de cerca por la Regresión por Vector de Soporte y los métodos basados en árboles.

Abstract

Formula 1 is considered to be the most prestigious and technological competition in the world of motorsport, where different teams and their drivers compete to win races and achieve the championship. By using historical data from previous races and applying prediction models based on Supervised Learning, it is possible to predict the results of the following races. This project aims to investigate the prediction of the championship winners, both drivers and constructors, by proposing different algorithms and following a data mining methodology.

A series of visualizations are presented to help understand the different variables of the dataset and the recent history of the sport. By preprocessing the data and testing the proposed algorithms, very encouraging results were achieved for the prediction of both championships. The best performing model was the Artificial Neural Network, closely followed by the Support Vector Regression and tree-based methods.

Palabras clave:

- Fórmula 1
- Visualizaciones
- Predicción
- Campeonato
- Regresión
- Aprendizaje Máquina Supervisado

Keywords:

- Formula 1
- Visualizations
- Prediction
- Championship
- Regression
- Supervised Learning

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introdución

En este primer capítulo se comentarán las motivaciones y los objetivos a conseguir mediante la realización de este proyecto.

1.1 Motivación

La Fórmula 1 (F1) es el deporte más avanzado tecnológicamente del mundo, y ahora más que nunca debido al auge del *Big Data*. Una gran cantidad de las decisiones y estrategias de carrera son tomadas en base a modelos predictivos que tienen en cuenta la diversidad de parámetros y escenarios que se dan a lo largo de las carreras.

Todos los deportes son impredecibles, pero la F1 tiene una serie de características que la hacen aún más imprevisible. Sucesos como los accidentes, roturas de motor, la climatología o un simple fallo en un cambio de ruedas no se pueden prever y alteran las predicciones de los modelos.

A nivel personal, siempre me ha surgido un gran interés acerca de este deporte, por lo que pienso que no hay mejor oportunidad que realizar el TFG aplicando a una pasión de la infancia los conocimientos que he adquirido a lo largo de este grado universitario.

1.2 Objetivos

La finalidad de este proyecto es recopilar y procesar datos para implementar un modelo de predicción que sea capaz de pronosticar los resultados de los pilotos en cada carrera. El objetivo final será predecir la clasificación del Campeonato de Pilotos y del Campeonato de Constructores con el mayor grado de precisión posible.

Esto se conseguirá haciendo uso de la información de las temporadas más recientes, y mediante la visualización de los datos y el procesamiento de estos, el desafío es comparar los resultados previstos por el modelo con los reales de la temporada 2022, aplicando técnicas de

Aprendizaje Máquina Supervisado (Supervised Learning (SL)).

La idea es utilizar únicamente los datos previos a la carrera, es decir, la información de dicha clasificación y de las carreras previas. Por ejemplo, para la décima carrera de la temporada, la información que se usará para predecir las posiciones de tal carrera será la clasificación de ese fin de semana y los resultados obtenidos en las nueve carreras disputadas. Esto se hace así ya que la información de la clasificación es muy importante para intentar predecir la posición de llegada, ya que puede ocurrir que la clasificación sea un poco extraña y que los pilotos se clasifiquen en posiciones muy distintas de las que quedarían en situaciones normales, afectando en gran medida a las predicciones.

Los objetivos concretos son:

- Analizar la información de las diferentes bases de datos y correlaciones entre variables mediante la creación de visualizaciones.
- Investigar la idoneidad y aplicabilidad de diferentes técnicas de SL para la predicción de los resultados.
- Comparar los resultados de los diversos modelos utilizados teniendo en cuenta diversas métricas.

1.3 Estructura de la memoria

La memoria está estructurada en 8 capítulos, que se describen a continuación:

- **Capítulo 1: Introdución**. Descripción de la motivación y objetivos del proyecto.
- **Capítulo 2: Descripción del dominio**. Se presentan los conceptos teóricos y el contexto del deporte en cuestión para familiarizar al lector con el problema a tratar. Para cada algoritmo se profundizará en su fundamento teórico, adjuntando una imagen del diagrama para entender su funcionamiento.
- **Capítulo 3: Metodología**. Explicación del método de trabajo adoptado, de la planificación estimada, el seguimiento real y los costes estimados asociados al proyecto.
- **Capítulo 4: Herramientas utilizadas**. Breve explicación de los medios y librerías empleadas para la realización del proyecto.
- **Capítulo 5: Estudio del conjunto de datos**. Extracción de nuevas características y análisis exploratorio de los datos a través de la visualización de los datos para su comprensión.

- **Capítulo 6: Modelado**. Incluye los procesos de preprocesado de los datos para su correcto funcionamiento con los modelos de predicción. También se introducen las métricas utilizadas.
- **Capítulo 7: Resultados**. Se presentan los resultados obtenidos con los distintos modelos sobre el conjunto de test. Se observa tanto el Campeonato de Pilotos como el Campeonato de Constructores. Se evalúan los resultados y se comentan las curiosidades observadas.
- **Capítulo 8: Conclusiones**. Se exponen las conclusiones que han sido extraídas, los conocimientos adquiridos y las posibles líneas futuras que se pueden seguir para proseguir el proyecto.

Capítulo 2

Descripción del dominio

En este capítulo se realizará una introducción a los dominios y conceptos, tanto del deporte como de los métodos utilizados, algo esencial para poder comprender el propósito y desarrollo del proyecto. Está dividido en tres secciones donde se comentará; primero, los conceptos de la Fórmula 1; segundo, el Aprendizaje Automático y los diferentes algoritmos disponibles; y por último, se presenta un análisis del estado del arte, analizando trabajos relacionados que han servido de inspiración para la realización de este proyecto.

2.1 Conceptos de la Fórmula 1

La F1 [1, 2] es la categoría más importante de las carreras de monoplazas. El primer campeonato mundial se celebró en 1950 y desde ese año la competición está regulada por la Federación Internacional de Automovilismo (FIA).

En los últimos años la competición está compuesta por 10 equipos, con dos pilotos por cada equipo. A lo largo de una temporada se compite por dos campeonatos: el Campeonato de Pilotos y el Campeonato de Constructores. Durante la temporada se disputan numerosos Grandes Premios (Grand Prix (GP)) en diversos circuitos repartidos por todo el mundo.

Un GP consiste en un fin de semana de carrera que se lleva a cabo de viernes a domingo. Primero hay una serie de Entrenamientos Libres, donde los equipos ajustan el coche y realizan diversas pruebas (neumáticos, tiempos de *pit stop* y otras estrategias de carrera). Luego tiene lugar la Clasificación, que se divide en 3 sesiones eliminatorias (Q1, Q2 y Q3) que sirve para determinar la parrilla de salida de la carrera, donde el piloto más rápido obtiene la *pole*, es decir, la primera posición de salida. Por último se produce la Carrera, el evento principal del GP, donde los pilotos y sus escuderías consiguen puntos en función de la posición obtenida en la carrera.

Además, hay muchos factores que alteran la posición de los pilotos en un GP: adelantamientos, paradas para cambio de neumáticos, accidentes, cambios en la climatología, etc.

2.2 Conceptos del Aprendizaje Automático

El Aprendizaje Máquina (Machine Learning (ML)) se trata de una rama de la Inteligencia Artificial que crea algoritmos que aprenden automáticamente a partir de los datos a través de un proceso de aprendizaje previo con el conjunto de entrenamiento. El algoritmo revisa los datos con el fin de encontrar unos patrones o reglas en el conjunto que le permitan hacer predicciones sin la necesidad de la intervención humana. Cuánto mayor sea la cantidad de datos que recibe el modelo, este tendrá una mayor capacidad de producir predicciones más precisas.

Existen diversos tipos de categorías de ML, pero destacan los siguientes:

- **Aprendizaje Máquina Supervisado (SL):** partimos de un conjunto de datos que ha sido etiquetado, es decir, se conoce el valor de la variable objetivo. Esto permite que el algoritmo aprenda una función capaz de predecir la variable para un nuevo conjunto de datos que no fue utilizado en el entrenamiento, proceso conocido como *"generalización"*. Dependiendo del tipo de la etiqueta, existen dos categorías.
  - **Regresión:** para predecir un valor numérico.
  - **Clasificación:** para predecir valores discretos, es decir, a qué clase pertenece cada individuo del conjunto de datos.
- **Aprendizaje Máquina No Supervisado:** se desconocen las clases de pertenencia de los individuos, sólo existen grupos similares. La *"no supervisión"* consiste en que es el sistema quien tiene que descubrir por sí solo las características de los datos de entrada.

En este trabajo se utilizarán únicamente métodos pertenecientes al SL. Los diversos modelos utilizados se explicarán en los siguientes apartados.

2.2.1 Árboles de Decisión

Los Árboles de Decisión (Decision Tree (DT)) [3, 4] son un tipo de algoritmo muy utilizado en el SL, ya que se trata de un modelo que se puede usar tanto para tareas de clasificación como de regresión. Está basado como su nombre indica, en un árbol, con una estructura de ramificaciones que muestran las decisiones que toma el modelo y cuya lectura se realiza de arriba a abajo. En un DT, cada nodo indica una variable del conjunto de datos y cada rama representa el valor que pueda tomar esa variable en concreto.

Hay varios tipos de nodos según la jerarquía:

- **Nodo raíz:** nodo inicial que indica la variable que mejor divide el conjunto de datos.
- **Nodos internos:** representan las variables que provienen de las sucesivas divisiones.

- **Nodos hoja:** representan todos los resultados posibles dentro del conjunto de datos, es decir, las clases si es de clasificación o los valores numéricos en caso de que sea un árbol de regresión.

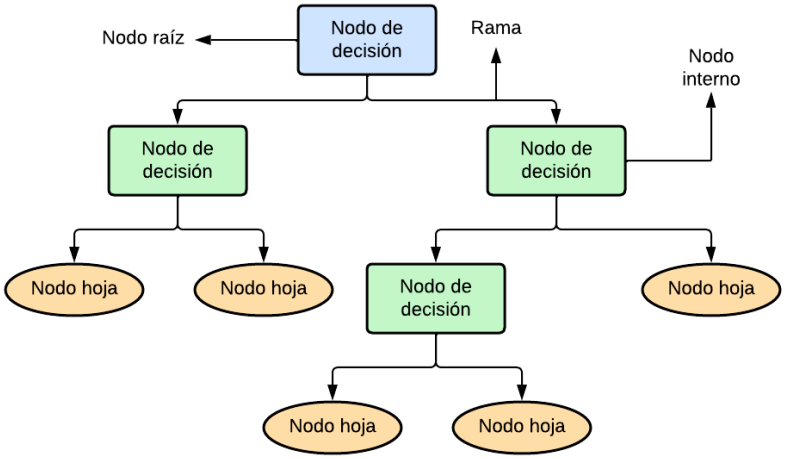


Figura 2.1: Diagrama de un árbol de decisión

El aprendizaje del DT emplea el algoritmo de Hunt para identificar los puntos de división óptimos de los datos. Este proceso de división se repite recursivamente hasta que la mayoría de los individuos se clasifiquen correctamente. La principal ventaja es la facilidad de comprensión del modelo, ya que la naturaleza jerárquica de un árbol de decisión también facilita ver qué atributos son los más importantes. El mayor problema es el posible sobreajuste, no siendo capaz de generalizar correctamente nuevos datos, inconveniente que se puede solucionar con procesos de poda (haciendo el árbol más sencillo, eliminando las divisiones que menos contribuyen y dejando los nodos más importantes).

2.2.2 Bosques Aleatorios

Los Bosques Aleatorios (Random Forest (RF)) [5] se basan en el concepto de *ensemble* [6]. Cada modelo produce una predicción distinta y estas se combinan para obtener una única predicción, siguiendo el lema de *"dos cabezas piensan mejor que una"*. La principal ventaja de esta combinación de modelos es que los errores tienden a compensarse ya que cada uno de ellos funciona de forma diferente, consiguiendo así un menor error de generalización final.

A la hora de hablar de algoritmos *ensemble*, hay distintas técnicas para combinar los resultados de los modelos, destacando los siguientes:

- ***Bagging*:** también conocido como *Bootstrap AGGregatING*, crea modelos diversos de forma paralela sobre distintas muestras aleatorias del conjunto original y luego combi-

na estos modelos en un resultado final mediante voto mayoritario o promedio [7]. Los subconjuntos se escogen uniformemente y con reemplazo (puede haber muestras duplicadas y muestras que no aparezcan nunca). Se trata de un método que busca reducir la varianza del modelo final. Son robustos frente al problema del sobreajuste de los datos.

- ***Boosting*:** es una secuencia de clasificadores, donde cada modelo intenta arreglar los errores realizados por los modelos previos. El primer modelo de la secuencia intentará aprender la relación entre las variables de entrada y la salida deseada, cometiendo algunos errores en la predicción. El siguiente modelo de la cadena tiene el objetivo de reducir esos errores. Este proceso se sigue iterativamente con los modelos, dando más peso a las muestras que han sido mal clasificadas y menor peso a las muestras correctamente clasificadas [8]. El algoritmo más conocido es *AdaBoost* (Adaptive Boosting (AB)). Se trata de una técnica de reducción tanto del sesgo como de la varianza y es más propenso al sobreajuste de los datos.

Los RFs son tipos especiales de *Bagging*, que se usan en combinación con árboles de decisión, que a la vez se construyen con subconjuntos aleatorios de características. Así se introduce más diversidad en el modelo y se reduce el tiempo de entrenamiento para cada árbol. RF es muy popular en tareas de clasificación y regresión, que construye varios árboles de decisión y calcula la salida como la moda de la clase predicha (clasificación) o la media (regresión).

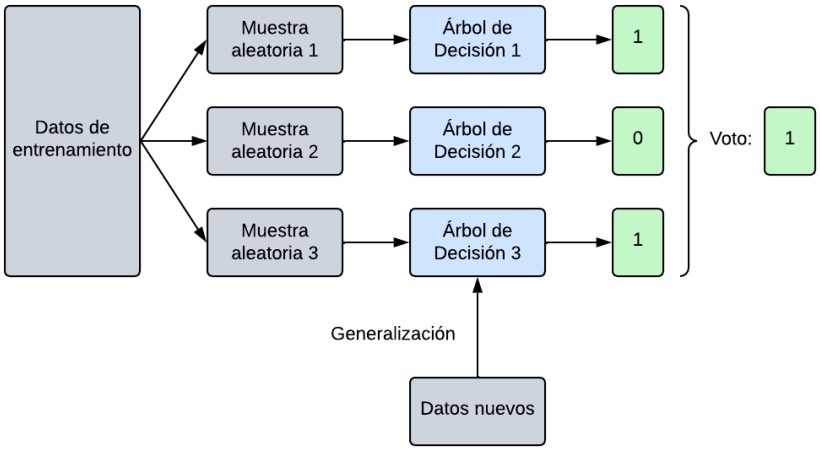


Figura 2.2: Diagrama de un árbol aleatorio

2.2.3 XGBoost

El Gradient Boosting (GB) se puede ver como un caso especial de AB, donde la principal diferencia es cómo los dos algoritmos identifican los problemas de los modelos débiles (árboles

de decisión). AB identifica los puntos problemáticos asignándoles pesos elevados mientras que GB lo hace a través del uso de gradientes en la función de pérdida [9].

En concreto, *XGBoost* (Extreme Gradient Boosting (XGB)) [10, 11] es una variante de GB que consigue muy buenos resultados y es altamente paralelizable. XGB usa una formalización del modelo más regularizada para controlar el sobreajuste, lo que resulta en mejores resultados.

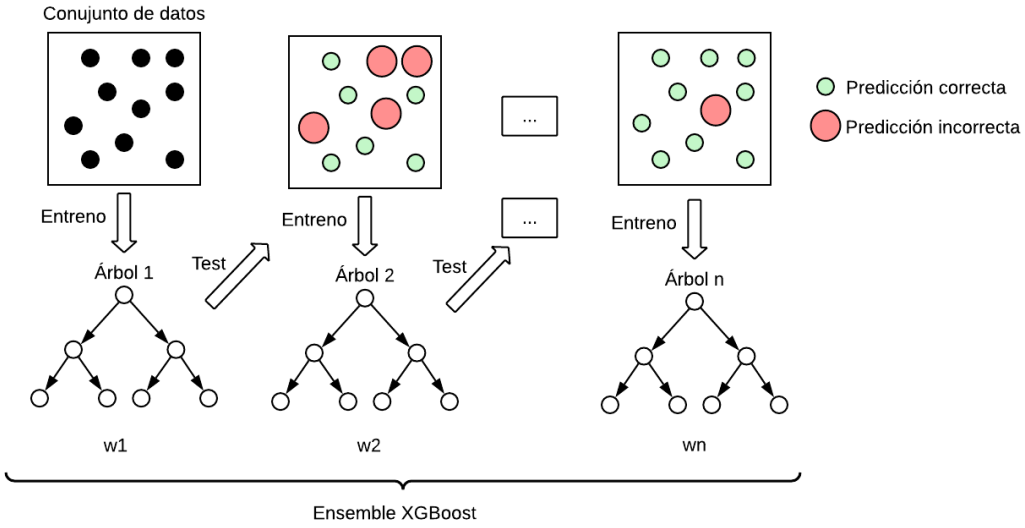


Figura 2.3: Diagrama de un Extreme Gradient Boosting

2.2.4 Máquinas de Vectores de Soporte

Las Máquinas de Vectores de Soporte (Support Vector Machine (SVM)) [12] son un algoritmo basado en el uso de hiperplanos a forma de separadores entre dos clases. El objetivo es encontrar el hiperplano óptimo, para el cual el margen entre los puntos de entrenamiento de dos diferentes clases es maximizado, de forma que el hiperplano que tenga el mayor margen es el mejor clasificador de los datos. Los patrones más cercanos al hiperplano se llaman *"vectores de soporte"* y solo estos importan, el resto de ejemplo de entrenamiento se pueden ignorar.

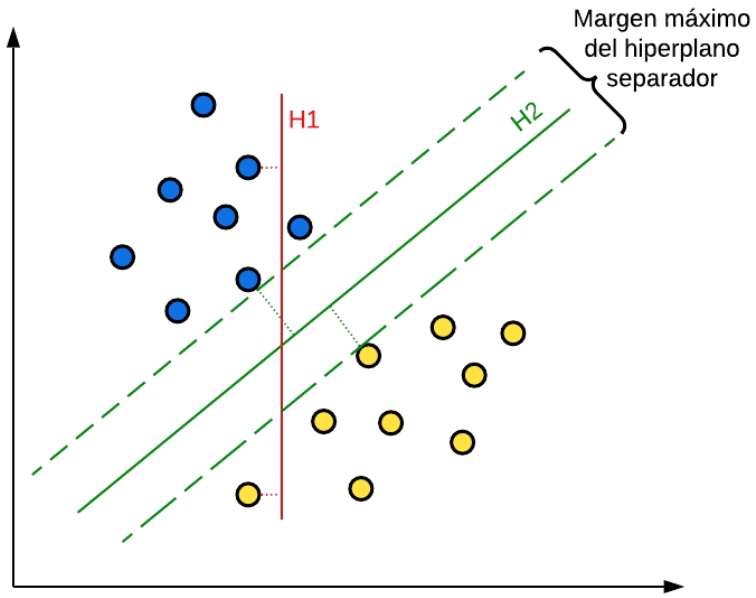


Figura 2.4: Diagrama de una máquina vector de soporte

En la figura 2.4 se observa un conjunto de datos con dos clases de datos (puntos azules y amarillos) y dos posibles hiperplanos (H1 y H2). El hiperplano separador H2 tiene un margen mayor entre las dos clases que H1, por lo que H2 tiene mayor capacidad de clasificar los datos y es escogido como el hiperplano óptimo.

En este ejemplo, las clases de datos son linealmente separables, pero no siempre es así. En caso de tener un problema no linealmente separable, se puede aplicar el *"Truco del kernel"*, que consiste en aplicar una función *kernel* a los datos para asignarlos a un espacio de mayor dimensionalidad donde los datos son separables. Hay varios tipos de funciones *kernel*: lineal, polinómico, función de base radial (RBF), sigmoide, etc.

En un principio, los SVMs son clasificadores entre dos clases, pero son generalizables a problemas multiclase mediante la estrategia *"uno contra todos"*. Además, también se pueden aplicar en problemas de regresión, conocido como Support Vector Regressor (SVR) [13], donde el objetivo es que los datos queden dentro del margen, a diferencia de lo que sucede en clasificación.

2.2.5 Redes Neuronales

Una Red de Neuronas Artificiales (RNA) es un modelo que se inspira en el funcionamiento del cerebro humano a la hora de procesar la información. Al igual que nuestro cerebro, una

RNA está compuesta por neuronas interconectadas entre sí, conocidas como neuronas artificiales que se agrupan en capas jerárquicas para procesar las señales. Los datos de entrada pasan por diversas capas hasta obtener la salida de la red. Cada nodo, o neurona artificial, se conecta a otro y tiene un peso y un umbral asociados. Hay varios tipos de capas [14]:

- **Capa de entrada:** es la capa que recibe los datos de entrada en la red neuronal. Cada neurona de esta capa representa una característica o variable de entrada.
- **Capa oculta:** esta capa procesa los datos de entrada y realiza cálculos matemáticos para generar nuevas características. Puede haber varias capas ocultas en una red neuronal.
- **Capa de salida:** produce el resultado final de la red neuronal. Cada neurona de esta capa representa una posible salida o categoría.

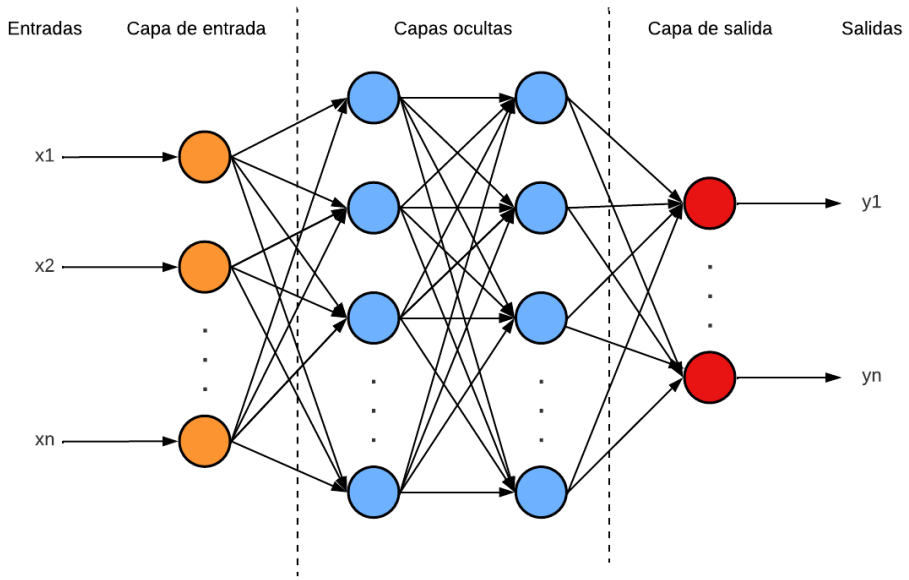


Figura 2.5: Diagrama de una red de neuronas artificiales

La figura 2.5 muestra el esquema de una RNA para clasificación. Para el caso de regresión, la capa de salida en una RNA de regresión tiene una sola neurona, que produce la predicción numérica.

Las neuronas se conectan entre sí mediante unos pesos, totalmente aleatorios al principio del proceso de entrenamiento, pero que se van modificando en un proceso iterativo conocido como aprendizaje y que consta de tres fases principales [15]:

- **Propagación hacia adelante:** la red recibe un conjunto de datos de entrada y se procesan mediante varias capas de neuronas interconectadas. En cada neurona se calcula

una suma ponderada de las entradas recibidas y el resultado se pasa a través de una función de activación para producir su salida asociada. Este proceso de propagación hacia adelante se repite hasta que se produce una salida al final de la red.

- **Cálculo del error:** una vez que se tiene la salida producida por la red, se compara con la salida esperada para calcular el error obtenido. Este error se mide usando una función de coste, que cuantifica la diferencia entre la salida esperada y la salida producida por la red. Existen diversas funciones de coste: error cuadrático medio (regresión), entropía cruzada (clasificación binaria), entropía cruzada categórica (clasificación multiclase), etc.

- **Propagación hacia atrás:** Se propaga el error hacia atrás a través de la red para ajustar los pesos y umbrales de las neuronas en un proceso llamado retropropagación del error (*backpropagation error*). El objetivo del entrenamiento es actualizar los pesos mediante un método iterativo basado en la información del gradiente de la función de error hasta llegar a su mínimo. Este proceso se repite muchas veces durante el entrenamiento de la red hasta que se minimiza el error en el conjunto de entrenamiento.

El diseño de la estructura de capas de neuronas de una red neuronal depende en gran medida del problema que se desea resolver. A menudo, el proceso de diseño se realiza de forma empírica, lo que significa que se prueban varias configuraciones hasta encontrar la que mejor se adapte a los datos y al problema. Depende de factores como el tipo de tarea, la complejidad del problema y la cantidad de datos. Las RNAs tienen varias ventajas y desventajas que deben ser consideradas. Las principales virtudes son:

- **Capacidad de modelado no lineal:** son capaces de modelar relaciones no lineales entre variables, siendo útiles en una amplia variedad de aplicaciones.

- **Adaptabilidad:** pueden adaptarse a nuevos datos y actualizar sus parámetros de forma que el modelo pueda mejorar con el tiempo.

- **Tolerancia al ruido:** son capaces de manejar datos con ruido y errores, siendo modelos beneficiosos en situaciones donde los datos pueden ser imperfectos.

- **Procesamiento en paralelo:** se pueden utilizar para procesar múltiples entradas a la vez, lo que las hace útiles en aplicaciones que requieren un procesamiento rápido.

Por otro lado, sus grandes inconvenientes son:

- **Requieren grandes cantidades de datos:** necesitan grandes cantidades de datos para entrenarse correctamente, lo que puede ser costoso y consumir mucho tiempo.

- **Necesitan mucho poder de cómputo:** son muy intensivas en cuanto al poder de procesamiento. Tiempos elevados para entrenar los modelos y necesidad de usar un hardware potente.
- **No son fácilmente interpretables:** son modelos de caja negra, lo que significa que es difícil entender cómo se están realizando las predicciones. No es posible comprender la relación entre la salida y la entrada ni la importancia de las características en el modelo.
- **Pueden sufrir sobreajuste:** pueden ajustarse demasiado a los datos de entrenamiento, por lo que pueden tener un bajo rendimiento con datos nuevos y desconocidos.

2.3 Estado del arte

Existen numerosos trabajos científicos enfocados a los resultados deportivos (natación, carreras de caballos, ciclismo, etc.) y algunos pocos en concreto a la F1. Partiendo del objetivo del preprocesado, el trabajo de Ofoghi *et al.* [16] propone la conversión de los tiempos del formato HH:MM:SS a segundos puros y posteriormente calcular el diferencial de tiempo para cada atleta como la diferencia entre el tiempo de cada atleta y el tiempo más rápido obtenido en esa competición en concreto. Este procedimiento será utilizado para computar la diferencia de tiempos en cada carrera contra el piloto que obtuvo la *pole*, para así poder crear una variable que mida la dominancia de los monoplazas.

Dentro de la documentación sobre la F1, hay diversos artículos que buscan encontrar el porcentaje de importancia del piloto y del monoplaza en los resultados obtenidos. Eichenberger y Stadelmann [17] realizan una investigación para encontrar al mejor piloto de todos los tiempos, considerando desde el año 1950 hasta el 2006. El objetivo es evaluar el talento de los pilotos, separando la componente del rendimiento de su coche, ya que cuanto mejor es el monoplaza, mejores resultados se suelen obtener. Para lograr separar la capacidad del piloto y del monoplaza en cuestión se realiza mediante regresión lineal y teniendo en cuenta diversas variables categóricas. Las conclusiones obtenidas son que el mejor piloto del período de estudio fue Juan Manuel Fangio y no Michael Schumacher, que se encuentra en tercera posición del ranking.

Bell *et al.* [18] realizan un estudio del éxito en la F1 a través de un modelado multinivel que tiene en cuenta la variable del piloto, del equipo y de la temporada. Al igual que en el artículo anterior, el mejor piloto según el modelo es Juan Manuel Fangio. Lo novedoso de este trabajo es el estudio de la importancia del piloto y del equipo en los resultados, llegando a la conclusión de que la variable de la escudería supera significativamente a los efectos del conductor, concretamente en un 86%.

Van Kesteren y Bergkamp [19] también buscaron separar la calidad del piloto de la variable

del equipo, utilizando para ello un modelo de regresión multinivel bayesiano para modelar el éxito individual de cada individuo como la proporción de competidores superados en cada carrera. Utiliza información correspondiente a las temporadas 2014 - 2021, usando además información extra como la climatología en cada carrera y el tipo de circuito (permanente o urbano). Lo curioso es que obtiene unos resultados semejantes al artículo previo, pese a que utilizan datos de distintas épocas, obteniendo un intervalo de confianza que, en torno al 87% de la variabilidad de los puntos obtenidos, se corresponde con la variable del equipo, mientras que únicamente el 13% tiene que ver al piloto. Esto concuerda con la opinión de muchos pilotos como Nico Rosberg, que comentó que el 80% del éxito en la F1 puede atribuirse al coche y 20% al piloto [20].

Con respecto a la predicción de resultados deportivos, nos encontramos con algunos proyectos que abordan este tema en concreto sobre la F1. Stoppels [21] utiliza redes neuronales para predecir los resultados de carreras, en concreto, las primeras 17 carreras de la temporada 2016 son utilizadas para predecir las 4 últimas carreras de dicha temporada. Estos resultados son comparados con otro método como la regresión logística y la conclusión es que los resultados obtenidos con la red neuronal son mejores.

Nigro [22] realiza el procesado de recolección de los datos, visualizaciones de estos y modelado de las predicciones. Se basa en predecir la probabilidad de que un determinado piloto gane un GP y compararla con las probabilidades de las casas de apuestas. No predice las posiciones de cada piloto en la carrera, sino que busca predecir el ganador de cada carrera de la temporada. A tal efecto, la variable objetivo puede ser tratada tanto de regresión (ordenar los resultados de cada carrera y adjudicar el menor valor como ganador) como de clasificación (la variable es 1 para el ganador y 0 para el resto). Para ambos problemas utiliza diferentes modelos: regresión lineal, RFs, SVMs y RNAs. El mejor modelo fue la red neuronal en el caso de clasificación, acertando el ganador del 62% de las carreras de la temporada 2019.

George [23] desarrolla una metodología para predecir los resultados de la temporada 2020. Para cada carrera obtiene las posiciones predichas, que luego son transformadas a los puntos que obtendrían los pilotos. Por último realiza la agregación de los puntos a lo largo de la temporada y crea una tabla comparativa de los resultados predichos y reales. El mejor resultado obtenido fue usando un modelo XGB, logrando un valor R^2 sobre el conjunto de test de 0.960. El principal inconveniente de este proyecto es que utiliza una variable denominada *"filled splits"* que determina la diferencia de tiempo de cada piloto con respecto al ganador de la carrera, algo que sólo se puede saber al finalizar la carrera, por lo que no es algo realmente válido incluirlo en el modelo si el objetivo es predecir utilizando información previa a la carrera.

Capítulo 3

Metodología

En este capítulo se planteará el método de trabajo utilizado, la planificación inicial y el seguimiento del proyecto, así como los recursos utilizados y sus costes.

3.1 Método de trabajo adoptado

Se emplea la metodología Cross Industry Standard Process for Data Mining (CRISP-DM), que indica el ciclo de vida de un proyecto de minería de datos. El ciclo de vida consiste en seis fases, mostradas en la figura 3.1, donde la secuencia de ejecución no es estricta.

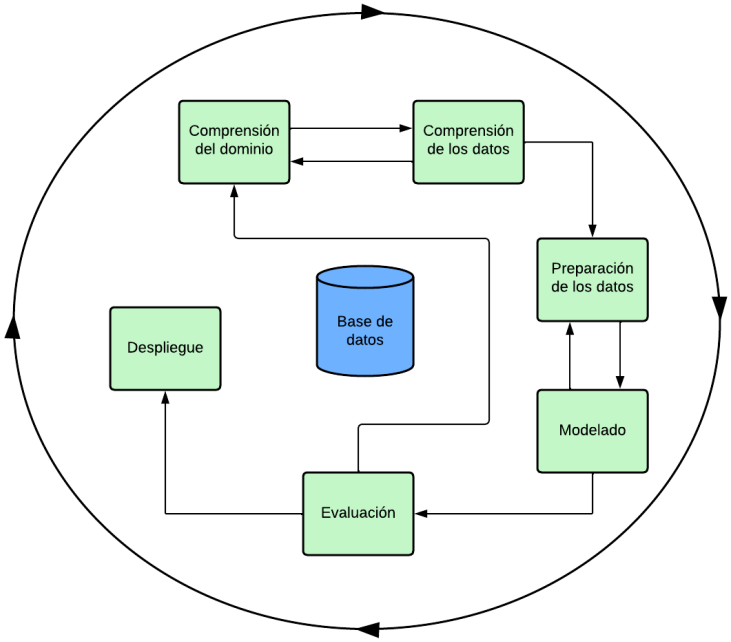


Figura 3.1: Fases del ciclo de CRISP-DM

De hecho, la mayoría de los proyectos avanzan y retroceden entre fases si es necesario. El proyecto no se termina una vez que la solución final se despliega, ya que se puede haber descubierto nueva información que puede producir mejores iteraciones en el modelo [24].

Bunker y Thabtah [25] desarrollaron una metodología basada en CRISP-DM pero adaptada al ámbito deportivo, denominada Sports Results Prediction CRISP-DM (SRP-CRISP-DM). Estas son las fases de la metodología propuesta:

1. **Comprensión del dominio:** comprensión del problema, del objetivo del modelado y de las características específicas del deporte en sí. Esto implica tener cierta comprensión del deporte en cuestión y qué factores podrían estar involucrados en determinar el resultado de las competiciones. Este conocimiento podría obtenerse a través del conocimiento personal del deporte, revisando la literatura existente o consultando a expertos en el deporte.

2. **Comprensión de los datos:** se recopila y se realiza un análisis exploratorio de los datos disponibles, incluyendo las estadísticas y estrategias de los equipos, los resultados anteriores y las condiciones meteorológicas. Se comprende la calidad, la relevancia y la adecuación de los datos para el problema que se debe resolver, razonando el tipo de variable objetivo (posiciones ordenadas o victoria) según el deporte.

3. **Preparación de los datos y Extracción de características:** son los procesos necesarios para crear el conjunto de datos que va a ser utilizado para entrenar el modelo de predicción. Se realiza la limpieza de datos, se integran diferentes fuentes de datos, se seleccionan y transforman las variables importantes (*"feature extraction"*) y se eliminan los datos redundantes.

4. **Modelado:** el objetivo es conseguir un modelo que cumpla con los requisitos iniciales del proyecto. Para ello es mejor probar diversas técnicas de ML, cuantas más mejor. Para cada modelo se ajustan los parámetros óptimos para conseguir los mejores resultados en un proceso llamado *"búsqueda de hiperparámetros"*. Posteriormente se realiza el entrenamiento y la validación.

5. **Evaluación:** antes de realizar el despliegue del modelo, es necesario saber la calidad de los modelos desarrollados, por lo que se lleva a cabo la evaluación sobre un conjunto que no ha sido utilizado durante el entrenamiento, conocido como conjunto de test.

6. **Despliegue:** se implementa el modelo en el entorno de producción. Se desarrollan planes para monitorizar y mantener el modelo a largo plazo y se realiza un seguimiento del rendimiento para asegurarse de que se está logrando el objetivo del modelo.

El flujo de trabajo a seguir para realizar el modelado de las predicciones sería el que se observa en la figura 3.2.

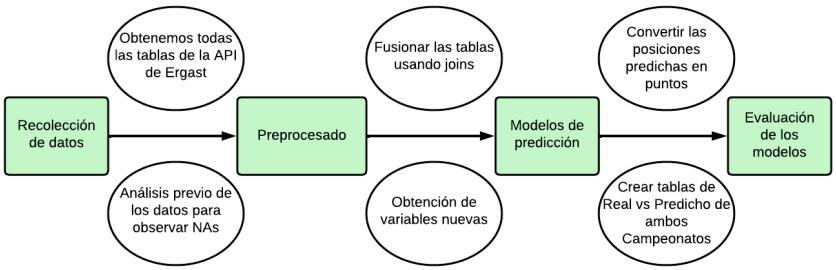


Figura 3.2: Flujo de trabajo

3.2 Planificación del proyecto

La planificación de este proyecto se divide en una serie de fases siguiendo la metodología explicada en el apartado anterior. Estas son las principales fases del trabajo:

1. **Anteproyecto y análisis bibliográfico del proyecto:** redacción del anteproyecto y búsqueda del estado del arte del proyecto, de las técnicas de preprocesado de datos y los diferentes modelos de ML.

2. **Recopilación de datos y procesado:** recopilar datos históricos, extraer nuevas características y crear visualizaciones para comprender los datos.

3. **Modelado de las predicciones:** preprocesado de los datos para el correcto funcionamiento de los modelos de predicción, implementación de los *pipelines* de características y de los procesos de búsqueda de hiperparámetros para cada modelo utilizado. Luego se realiza el entrenamiento del modelo.

4. **Evaluación de los resultados:** se utiliza el conjunto de test para crear las predicciones de los modelos. Se analizan los resultados y se crean unas tablas comparativas para facilitar la lectura de la información.

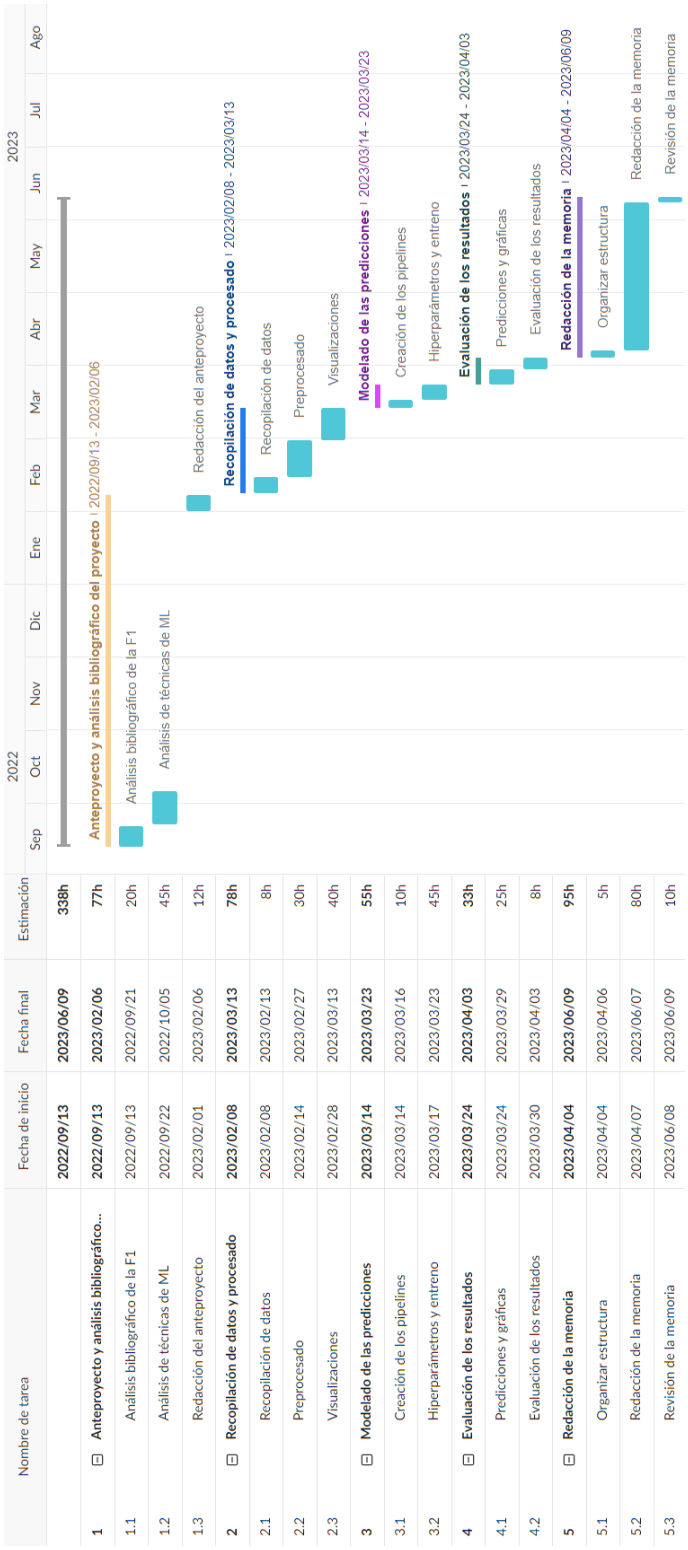
5. **Redacción de la memoria:** se procede a la redacción de la memoria y a la revisión final del proyecto.

La planificación prevista comienza a finales de junio de 2022, contactando con el tutor para indicarle el deseo de realizar el TFG sobre el tema de la F1. A mediados de septiembre de 2022, con el inicio del curso, comienza la búsqueda de bibliografía y de las bases de datos. Debido a la cantidad de asignaturas y las prácticas de empresa, optamos por proseguir con el

proyecto una vez terminado el primer cuatrimestre del curso. Según el plan acordado con el tutor, a inicios de febrero de 2023 se retomaría el plan establecido, redactando el anteproyecto, comenzando a programar código y creando los primeros entregables que se irían comentando periódicamente con el tutor. Como fecha de finalización del código, se estableció que a inicios de abril de 2023 debería estar terminado, para poder empezar con la redacción de la memoria y tenerlo todo listo para la convocatoria de junio.

3.3 Seguimiento del proyecto

No ha sido necesario recalcular las fechas previstas inicialmente. Cada una de las fases comentadas anteriormente está compuesta por tareas detalladas, por ejemplo *"1.1 Análisis bibliográfico de la F1"*. En la figura 3.3 se muestra la duración real del proyecto, con las fechas de inicio y fin de cada tarea, junto a la cantidad de horas de duración para cada una de ellas.



	Nombre de tarea	Fecha de inicio	Fecha final	Estimación
		2022/09/13	**2023/06/09**	**338h**
1	⊟ **Anteproyecto y análisis bibliográfico...**	**2022/09/13**	**2023/02/06**	**77h**
1.1	Análisis bibliográfico de la F1	2022/09/13	2022/09/21	20h
1.2	Análisis de técnicas de ML	2022/09/22	2022/10/05	45h
1.3	Redacción del anteproyecto	2023/02/01	2023/02/06	12h
2	⊟ **Recopilación de datos y procesado**	**2023/02/08**	**2023/03/13**	**78h**
2.1	Recopilación de datos	2023/02/08	2023/02/13	8h
2.2	Preprocesado	2023/02/14	2023/02/27	30h
2.3	Visualizaciones	2023/02/28	2023/03/13	40h
3	⊟ **Modelado de las predicciones**	**2023/03/14**	**2023/03/23**	**55h**
3.1	Creación de los pipelines	2023/03/14	2023/03/16	10h
3.2	Hiperparámetros y entreno	2023/03/17	2023/03/23	45h
4	⊟ **Evaluación de los resultados**	**2023/03/24**	**2023/04/03**	**33h**
4.1	Predicciones y gráficas	2023/03/24	2023/03/29	25h
4.2	Evaluación de los resultados	2023/03/30	2023/04/03	8h
5	⊟ **Redacción de la memoria**	**2023/04/04**	**2023/06/09**	**95h**
5.1	Organizar estructura	2023/04/04	2023/04/06	5h
5.2	Redacción de la memoria	2023/04/07	2023/06/07	80h
5.3	Revisión de la memoria	2023/06/08	2023/06/09	10h

Figura 3.3: Diagrama de Gantt de la duración del proyecto

El tiempo total que se ha empleado en la realización del proyecto han sido 338 horas. Las fases de análisis bibliográfico, modelado de las predicciones y redacción de la memoria son las que más tiempo han consumido para la realización del proyecto.

3.4 Costes del proyecto

Respecto a los costes estimados de los recursos humanos, se han consultado y analizado los datos de diversas *webs* donde miles de trabajadores comparten información de las condiciones de su puesto de trabajo. Se puede considerar que un ingeniero de datos recién graduado (*junior data engineer*) cobra unos 20€/h. Una vez realizado el cálculo de las horas empleadas en el proyecto, excluyendo la fase de redacción de la memoria, se realiza el siguiente cálculo:

Puesto trabajo	Tiempo	Coste	Total
Ingeniero de Datos Junior	243 horas	20€/h	4760€

Tabla 3.1: Estimación de costes de recursos humanos

Todas las herramientas de software utilizadas son de acceso gratuito, por lo que no suponen un coste adicional. No obstante, para llevar a cabo un proyecto se necesita el material de trabajo y servicios, que suponen unos gastos a mayores. Dejando de lado gastos como el acceso a internet o la electricidad, los costes de material son los siguientes:

Equipo	Coste
Ordenador personal	1000€
Tablet para consultar bibliografía	600€
Teclado y monitor	300€
Total	1900€

Tabla 3.2: Costes de recursos materiales

Si combinamos los costes de recursos humanos y los costes de materiales, el coste total estimado del proyecto resulta ser de 6760€.

Capítulo 4

Herramientas utilizadas

En este capítulo se comentará el lenguaje de programación y las principales librerías utilizadas, así como la forma de acceder a la base de datos.

4.1 Lenguaje de programación

Python [26] es un lenguaje de programación interpretado, de alto nivel y multiparadigma. Fue creado a finales de la década de 1980 y se ha convertido en uno de los lenguajes de programación más populares en la actualidad. Tiene una sintaxis legible y clara, y una gran cantidad de librerías disponibles. Es utilizado en una amplia variedad de aplicaciones, lo que lo hace una herramienta esencial para muchos programadores y científicos de datos. En concreto, en este trabajo utilizaremos la versión 3.9.7.

Como entorno de desarrollo se ha utilizado Visual Studio Code (VSCode) [27], un editor de código fuente popular, fácil de usar y personalizable, que es compatible con varios lenguajes de programación y tiene una amplia variedad de características y extensiones disponibles.

4.1.1 Principales librerías

A continuación se describen brevemente las principales librerías de Python que han sido utilizadas para el proyecto:

- **Pandas [28]:** librería especializada en el manejo y análisis de estructuras de datos. Permite reordenar, dividir, combinar conjuntos de datos, además de leer y escribir ficheros en el formato CSV.
- **Numpy [29]:** librería utilizada para el cálculo numérico y el análisis de datos, centrada en trabajos con un gran volumen de datos.
- **Requests [30]:** es una herramienta popular para realizar peticiones HTTP en Python. Proporciona una interfaz fácil de usar para enviar solicitudes, permitiendo a los de-

sarrolladores interactuar con servidores web y API en línea de una manera sencilla y eficiente.

- **Seaborn [31]:** librería de visualización de datos que se basa en la librería Matplotlib [32] y que proporciona una interfaz más fácil de usar para la creación de gráficos estadísticos atractivos. Seaborn incluye una amplia variedad de tipos de gráficos, desde gráficos de distribución y de correlación hasta gráficos de regresión y de series temporales.
- **Dython [33]:** librería pensada para el análisis de datos. Dado un conjunto de datos, Dython encuentra automáticamente qué características son categóricas y cuáles son numéricas, calculando una medida relevante de asociación entre cada una de las características, y lo representa todo como un mapa de calor fácil de leer.
- **Scikit-learn [34]:** librería de ML que proporciona herramientas para el análisis de datos y la construcción de modelos predictivos. La librería incluye una amplia gama de algoritmos de aprendizaje supervisado y no supervisado, así como herramientas para la selección de características, la evaluación de modelos y la validación cruzada.
- **xgboost [35]:** librería que proporciona la implementación de un XGB.

4.2 Base de datos

Los datos fueron recolectados a través de la API de Ergast [36], la cual contiene datos extraídos del sitio web de F1. La API en sí misma contiene toda la información sobre carreras, resultados, pilotos, clasificaciones, tiempos de vuelta, paradas en boxes, clasificaciones de constructores y pilotos, y circuitos desde los inicios de F1 hasta la última carrera disputada. A continuación, en la figura 4.1 se muestra el diagrama entidad - relación de la base de datos:

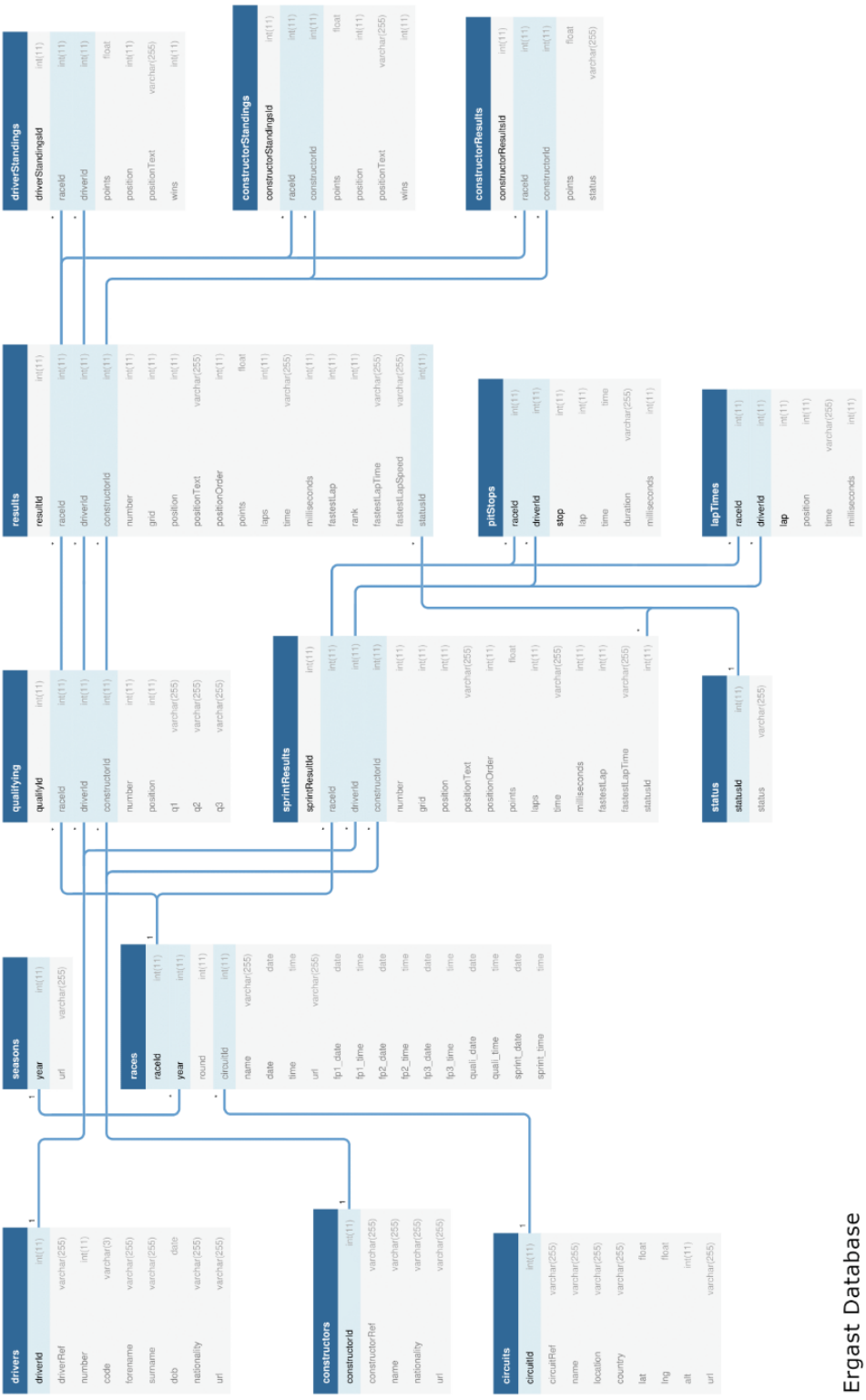


Figura 4.1: Modelo entidad - relación Ergast

Capítulo 5

Estudio del conjunto de datos

En este capítulo se llevará a cabo un proceso de extracción de nuevas características que tendrán como objetivo mejorar las predicciones de los modelos. También se mostrará la descripción de las diferentes variables del conjunto de datos. Por último, se crearán diferentes visualizaciones que servirán para comprender mejor los datos.

5.1 Extracción de características

Utilizando las variables obtenidas a través de Ergast, se ha realizado un tratamiento de los datos para crear una serie de nuevas variables que ayuden al modelo a crear mejores predicciones:

- Calcular la edad relativa de los pilotos en cada carrera. Esto es una medida que refleja la experiencia que tiene cada piloto.
- Calcular la diferencia de cada piloto con respecto al que obtuvo el tiempo más rápido en la clasificación. Representa la dominancia de los pilotos y los monoplazas, ya que no es lo mismo terminar a 1 décima que a 1 segundo. Como no todos los pilotos llegan a Q2 y Q3, se crea una lista de los tiempos para cada piloto y se escoge el tiempo más rápido que ha conseguido marcar. Luego se calcula el diferencial de tiempos entre cada piloto frente al tiempo de la *pole*.
- Se dice que la F1 es un deporte de rachas, que sólo importa lo que haya realizado el piloto en la última carrera. "*They say you're only as good as your last race so, although the Melbourne result was a great morale booster, we now have to start all over again in a race that will be a much tougher proposition*" [37]. A tal efecto, se calcula la media móvil de las últimas tres carreras y clasificaciones de cada piloto, mediante el uso de la función *"rolling"* que proporciona Pandas, procedimiento similar al que realiza Terenzio [38].

Estas variables sirven para mostrar la supremacía y constancia de rendimiento de las últimas clasificaciones y carreras.

- Tal y como se explicó en el apartado de objetivos (sección 1.2), la idea es utilizar información previa a las carreras. Por lo tanto, se crean variables que indican los puntos obtenidos por cada piloto y constructor antes de cada carrera. Al igual que las últimas variables creadas, sirve para indicar la hegemonía a lo largo de los campeonatos.

5.2 Visualizaciones

Para realizar un buen modelado es necesario utilizar unos datos relacionados entre sí, por lo que se tienen en cuenta los datos a partir de la temporada 2014, año en el que hubo un gran cambio de reglamento que sigue vigente en la temporada 2022 que se quiere predecir. Como resultado del proceso anterior de extracción de características, el conjunto final de variables está formado por las siguientes 13 características.

Variable	Descripción	Ejemplo
temporada	Año del campeonato	2022
num_carrera	Número de carrera	22
circuito	Nombre del circuito	yas_marina
piloto	Nombre del piloto	alonso
constructor	Nombre del equipo	alpine
pos_parilla	Posición de la parrilla de salida	10
dif_clasificacion	Dif. con respecto al tiempo de pole	1.272
ult_3_carreras	Dif. media últimas carreras	50.804
ult_3_clasificaciones	Dif. media últimas clasificaciones	0.837
puntos_previos_piloto	Puntos del piloto hasta esa carrera	81
puntos_previos_const	Puntos del equipo hasta esa carrera	167
edad_piloto	Edad relativa del piloto en días	15089
pos_final	Posición final llegada en la carrera	20

Tabla 5.1: Descripción de las variables del conjunto de datos

Usando estos datos, se pueden realizar distintas visualizaciones para descubrir información a mayores que permita comprender las diferentes variables, su importancia y cómo se fue desarrollando la F1 en las últimas temporadas.

5.2.1 Correlación de las variables

Una matriz de correlaciones [39] indica la relación entre varias variables en un conjunto de datos. Es una matriz cuadrada que muestra el coeficiente de correlación entre pares de variables. Los valores en la diagonal principal de la matriz son siempre 1, ya que cada variable está completamente correlacionada consigo misma. Los valores fuera de la diagonal principal muestran el grado de correlación entre dos variables. La matriz de correlaciones es útil para explorar la estructura de los datos y comprender cómo las variables están relacionadas entre sí. Puede ayudar a identificar posibles relaciones entre variables y ayudar en el proceso de selección de características en los modelos de ML.

En nuestro conjunto de datos tenemos tanto variables numéricas como categóricas, por lo que sería interesante observar las correlaciones entre ambos tipos de variables. Las medidas de correlación tradicionales como el coeficiente de correlación de Pearson no son adecuadas para este propósito, ya que están diseñadas para medir la correlación entre dos variables numéricas. No obstante, sí es posible calcular correlaciones entre variables numéricas y categóricas usando las técnicas adecuadas [40, 41]. A continuación se muestran las técnicas utilizadas para comparar correlaciones entre distintos tipos de pares de variables:

- **Numérica y numérica**: se usa la *"correlación de Pearson"*. Su valor puede oscilar entre -1 y 1, donde un valor de 1 indica una correlación positiva perfecta, un valor de -1 indica una correlación negativa perfecta y un valor de 0 indica que no hay correlación lineal.
- **Categórica y categórica**: la *"correlación de Cramer's V"* es una medida de asociación entre dos variables categóricas que se basa en la prueba chi-cuadrado. Su valor va de 0 a 1, donde 0 supone la ausencia de asociación y 1 es una asociación perfecta. Otra opción similar es la "*correlación de Theil's U*".
- **Categórica y numérica**: se utiliza la *"razón de correlación"* (*correlation ratio*), cuyo valor también oscila entre 0 y 1. Es una medida de la relación curvilínea entre la dispersión estadística dentro de las categorías individuales y la dispersión en toda la población o muestra.

Todos estos casos están contemplados y correctamente implementados en la función *"associations"* de la librería Dython, comentada en la sección de librerías utilizadas 4.1.1. Aplicando esta función al conjunto de datos, se obtiene la matriz de correlaciones que se observa a continuación en la figura 5.1.

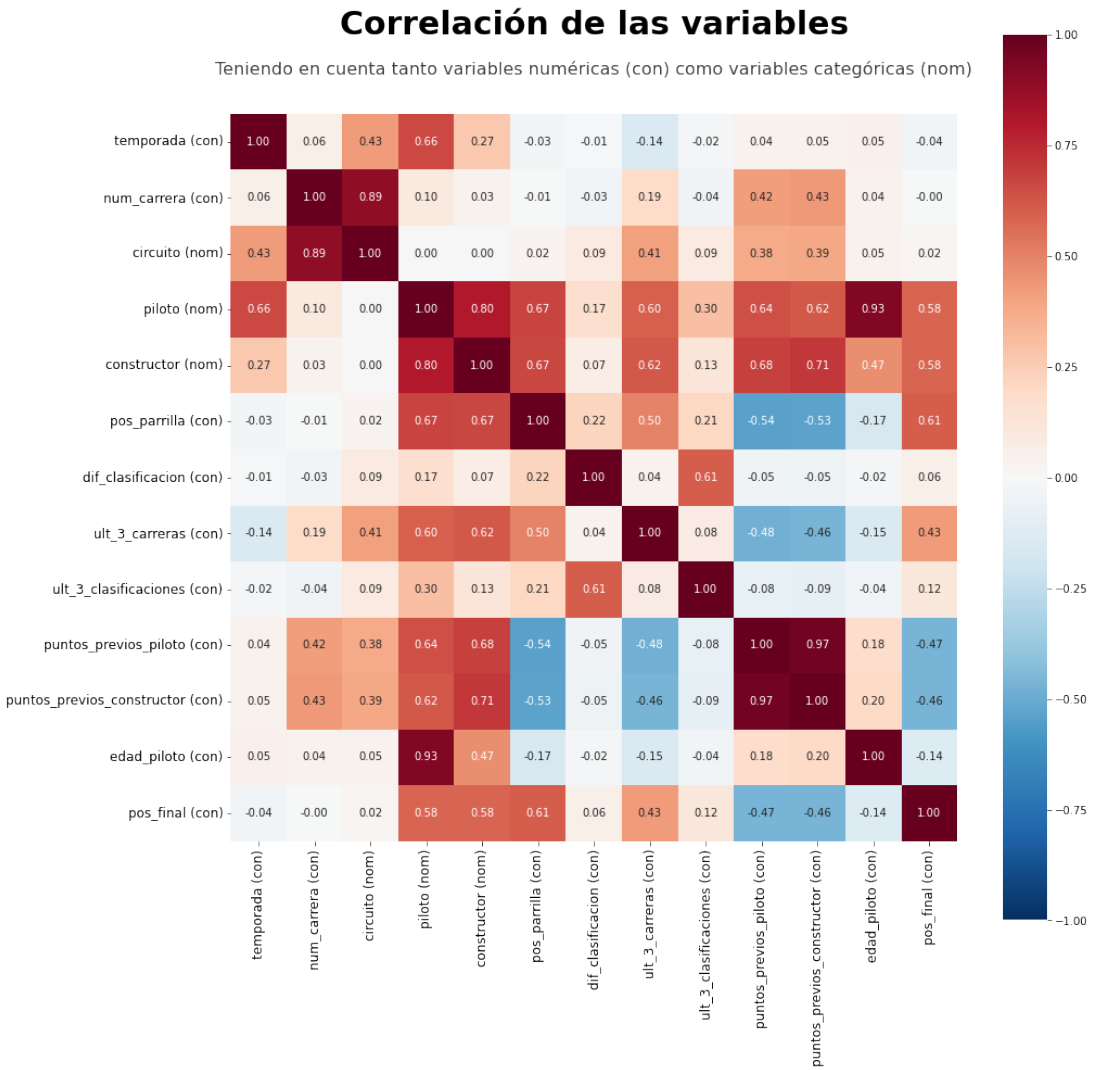


Figura 5.1: Correlaciones de las variables

La variable a predecir es la posición final (*"pos_final"*), por lo que nos interesa conocer aquellas características más relacionadas. Con un valor de 0.61, *"pos_parrilla"* es la variable con mayor correlación, indicando que la posición de salida y la posición de llegada suelen estar bastante relacionadas entre sí. Sin embargo, esta correlación no es perfecta y pueden haber factores externos que influyan en el resultado final de la carrera, como accidentes, fallos mecánicos, sanciones, cambios climáticos, entre otros. Además, en algunas pistas es más difícil adelantar a otros coches y esto puede influir en la correlación entre la posición de salida y la de llegada. En general, se considera que la posición de salida es un buen predictor del resultado final de la carrera, pero no es determinante. Luego, las variables *"piloto"* y *"constructor"* tienen un valor de 0.58, confirmando que la calidad de los monoplazas explican gran parte del

resultado obtenido. Después, con un coeficiente de 0.43 está la variable *"ult_3_carreras"*, por lo que la racha que tienen los pilotos en las últimas carreras disputadas explica en buena parte cómo van a realizar la carrera. Por último, con valores negativos de correlación están las variables que muestran los puntos previos obtenidos por el piloto y por su equipo. Estos valores son negativos ya que a mayor cantidad de puntos, más posibilidades tienen de colocarse en las primeras posiciones de llegada.

5.2.2 Ratio de conversión de *poles* en victorias

El ratio de conversión de *poles* en victorias según el circuito puede variar significativamente según la pista en cuestión [42]. Algunos circuitos tienen una correlación más fuerte entre la *pole* y la victoria, mientras que en otros la correlación es más débil. Por ejemplo, el circuito de Mónaco es conocido por ser estrecho y difícil de adelantar, mientras que el circuito de Spa-Francorchamps tiene una pista más larga y ancha donde los adelantamientos son más frecuentes. En la figura 5.2 se observan los circuitos ordenados según el ratio de conversión de *poles* en victorias.

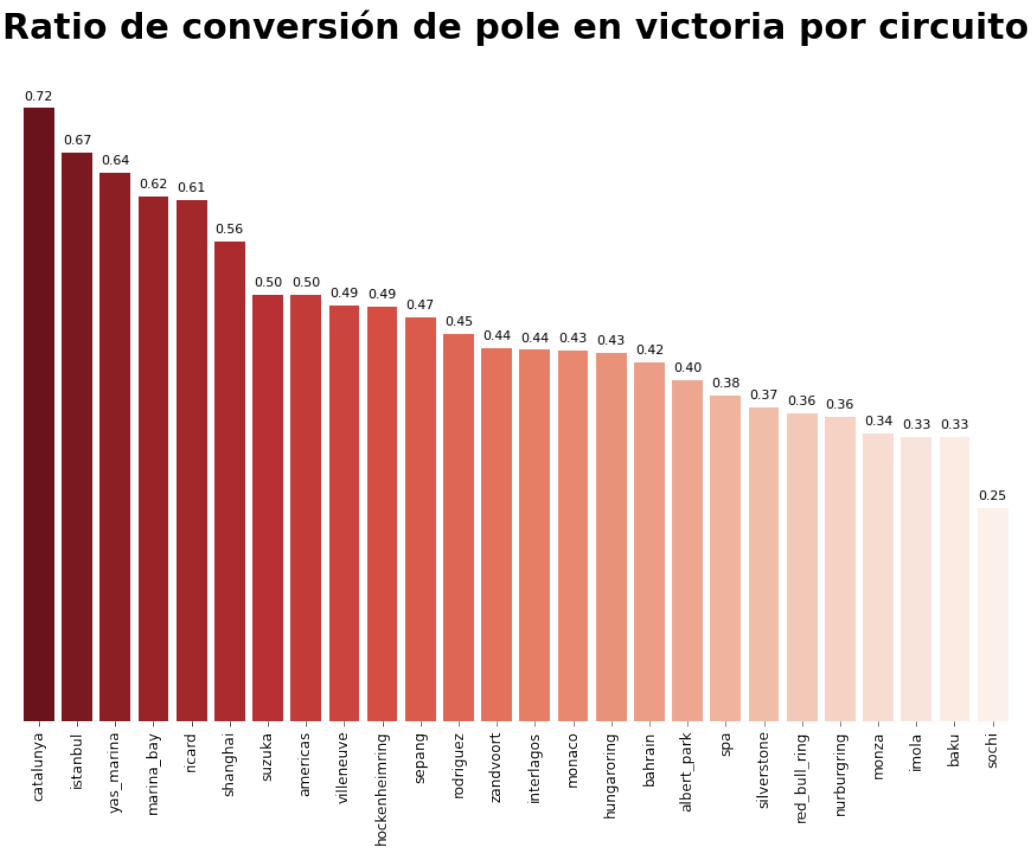


Figura 5.2: Ratio de conversión de poles

Resulta ser que el circuito de Cataluña es el que tiene un mayor ratio, con un 72%, siendo de gran importancia obtener la *pole* para intentar asegurar la victoria. Esto se debe a que los tests de pretemporada se suelen realizar en esta pista, por lo que los equipos tienen mucha información para optimizar al máximo sus coches. Además, es un circuito calificado como aburrido, ya que cuenta con pocas rectas y con pocos puntos de adelantamiento.

Sorprende Mónaco, con un 43%, ya que los adelantamientos son muy difíciles y en teoría el porcentaje de conversión debería ser más alto. Esto se explica observando el circuito, siendo de los lugares más reñidos para pilotar y demostrar la calidad al volante. Los pilotos deben estar concentrados durante toda la carrera, ya que cualquier despiste supone no terminar la carrera. En esta pista, como no es fácil adelantar, las estrategias son más relevantes para lograr avanzar posiciones. Aquí se aplican dos conceptos de estrategia sobre el cambio de ruedas:

- ***Overcut:*** el piloto se queda en la pista más tiempo que sus rivales antes de hacer una parada en boxes. La idea es que, al conducir con neumáticos más desgastados, el piloto puede hacer un tiempo de vuelta más rápido que sus rivales que ya han parado en boxes y tienen neumáticos nuevos.
- ***Undercut:*** es lo contrario al overcut. En esta estrategia, el piloto hace una parada en boxes antes que sus rivales para poner neumáticos más nuevos y tratar de ganar tiempo en la pista mientras sus rivales todavía tienen neumáticos viejos y gastados.

5.2.3 Legado de los equipos desde el comienzo de la era híbrida

Un cambio de reglamento en la F1 puede tener una serie de consecuencias, tanto positivas como negativas, para los equipos y los pilotos. Permite que los equipos que estaban en desventaja puedan mejorar su rendimiento, mientras que los equipos que estaban en la cima pueden sufrir, pero también puede resaltar el poder y reinado de algún equipo y convertir la competición en algo aburrido y previsible.

El cambio de reglamento en 2014 [43] fue uno de los más importantes en los últimos años. En primer lugar, se introdujeron nuevos motores V6 turbo híbridos, en lugar de los antiguos motores V8 atmosféricos. Estos nuevos motores eran mucho más eficientes en términos de combustible, pero también más complicados y costosos de fabricar. Además de los nuevos motores, también se introdujeron cambios en los sistemas de recuperación de energía. Estos cambios inauguraron la conocida *"era híbrida"*.

A lo largo de los años, algunos de los equipos han cambiado de nombre, siendo los casos más recientes el de Renault a Alpine y Racing Point a Aston Martin. El nombre del constructor cambia, pero la estructura del equipo y la fábrica se mantiene, por lo que se opta por unificar los equipos bajo el último nombre registrado, para así poder observar la tendencia a lo largo

de los últimos años. En la figura 5.3 se observan los puntos obtenidos por cada equipo en cada temporada, permitiendo observar tendencias y rendimientos a lo largo de los años.

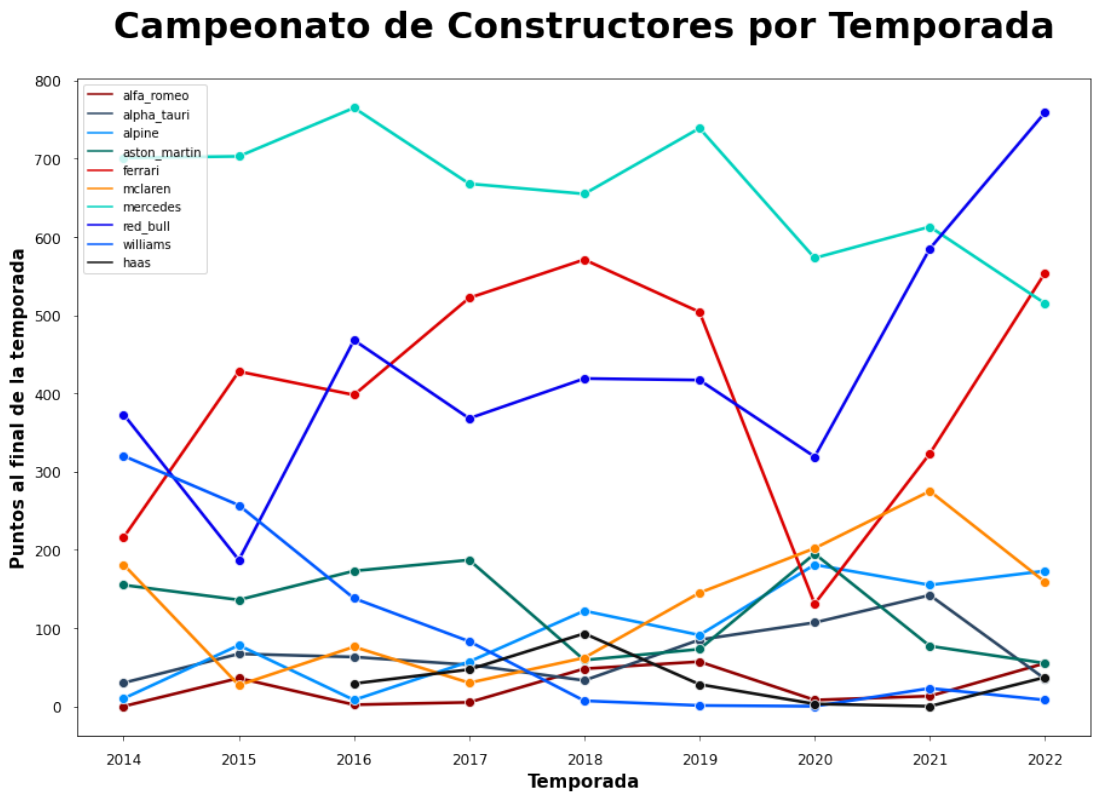


Figura 5.3: Resultados históricos por constructor

Al comienzo de este reglamento, el equipo Mercedes dominó por un gran margen sobre el resto de equipos. A lo largo de las siguientes temporadas, Ferrari y Red Bull fueron acercándose al campeón, alternando entre segunda y tercera posición, pero sin ser capaces de ganar el Campeonato de Constructores. Destaca la caída de rendimiento de Ferrari en 2020, culpa de una penalización de la FIA sobre la legalidad del motor Ferrari en la temporada previa. El resto de equipos, salvo alguna temporada aislada, no pudieron ni acercarse al pretendiente al título.

Para intentar mejorar el espectáculo y equilibrar la competición, en el año 2022 se introdujo un nuevo reglamento [44], el *"efecto suelo"*. Las nuevas normas para la temporada 2022 presentan una serie de cambios significativos en la aerodinámica de los coches, que tienen como objetivo reducir la turbulencia y permitir una mayor cercanía entre los monoplazas en pista. Además, se introduce un límite presupuestario para los equipos, con el objetivo de reducir la brecha entre los equipos grandes y los más pequeños. Con el nuevo reglamento, Red Bull le arrebató el campeonato a Mercedes y Ferrari recuperó parte de su competitividad.

5.2.4 Media de las vueltas rápidas por temporada

Al inicio de un reglamento, los equipos no tienen mucha información sobre los monoplazas, ya que muchos de los componentes son nuevos y no se han probado lo suficiente. Los equipos siempre están buscando maneras de mejorar su rendimiento en pista, y esto implica un constante trabajo de desarrollo y evolución de sus coches. Cuando hay un reglamento establecido, los equipos pueden trabajar en ajustar y mejorar sus coches dentro de los límites establecidos por el mismo. Aspectos como mejorar la aerodinámica del coche y crear nuevos componentes son la base del desarrollo de un monoplaza. Con el tiempo, estos avances pueden llevar a mejoras significativas en los tiempos por vuelta, y pueden permitir que los equipos luchen por los primeros puestos en las carreras.

Para hacer un estudio de los avances de los tiempos por vuelta a lo largo de las temporadas, la comparativa debe realizarse teniendo en cuenta las mismas características, es decir, observando los tiempos sobre los mismos circuitos. Para ello se analizan únicamente aquellos circuitos donde se han disputado carreras en todas las últimas temporadas contempladas, teniendo una lista de 10 pistas, con los circuitos de Cataluña, Mónaco, Bahrain, Silverstone y Monza entre otros.

Para cada carrera se calcula la mediana de las vueltas rápidas de todos los pilotos, resultando en un único valor que nos permite hacer comparaciones. Para una temporada en concreto se tiene una lista de 10 valores, que se convierten en un diagrama de cajas. Se realiza el mismo proceso para todas las temporadas, tal y como se observa en la figura 5.4.

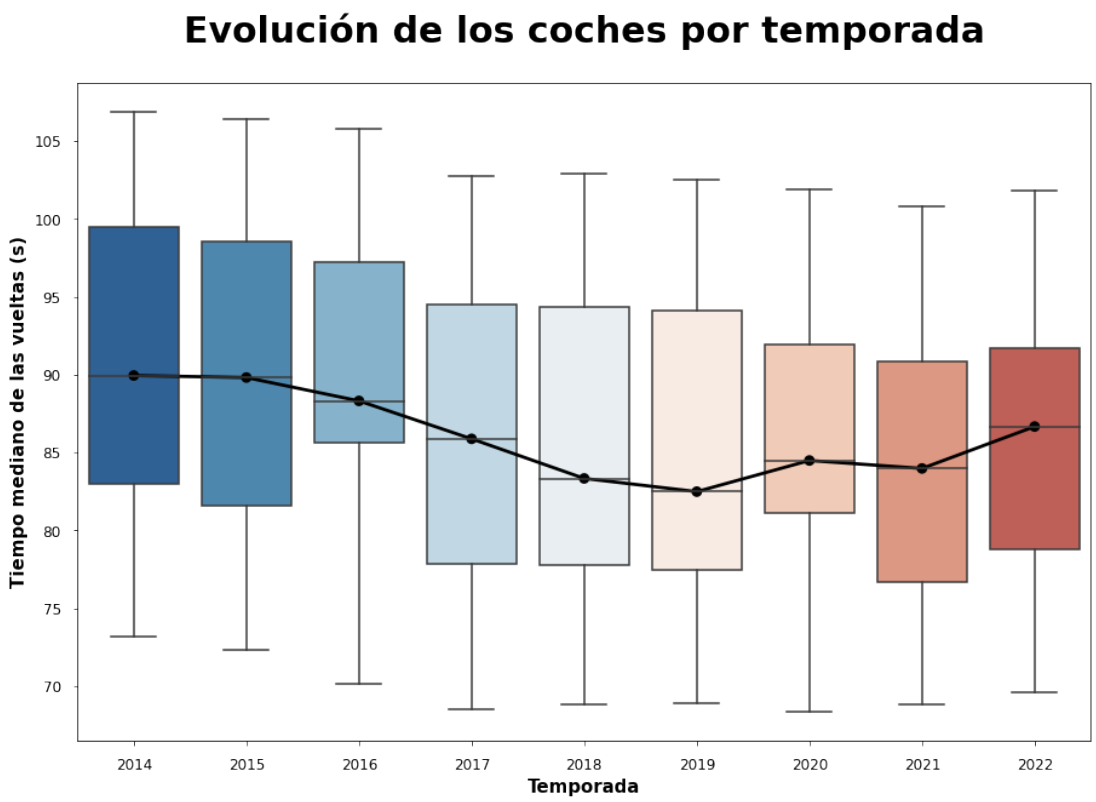


Figura 5.4: Evolución de los coches por temporada

A partir del 2015, los equipos fueron desarrollando mejoras, haciendo que los tiempos por vuelta fuesen más bajos, es decir, los monoplazas eran más rápidos. Esta tendencia se sucede hasta el año 2019, donde se introdujo un ligero cambio en el reglamento. El cambio más observable es el aumento de tiempos del 2021 a 2022, debido al nuevo reglamento explicado anteriormente.

Capítulo 6

Modelado

En este capítulo se explicarán las decisiones tomadas con respecto al proceso de modelado de las predicciones. Se tratarán aspectos como el tipo de problema planteado, las métricas utilizadas para evaluar la bondad de las predicciones, y el procesado realizado sobre los datos.

6.1 ¿Regresión o clasificación?

El objetivo es predecir las posiciones de las carreras, y tal como se explicó en la sección 2.2, hay dos tipos de problemas dentro del SL. Por un lado, las posiciones son valores discretos (1º, 2º, 3º, etc.) y en este caso se realiza una clasificación multiclase, donde cada posición sería una clase distinta. El problema es que existe el peligro de que el modelo clasifique para la misma clase (posición de carrera) a varios pilotos y que haya posiciones no asignadas. No se puede establecer que los modelos de clasificación seleccionen únicamente un individuo para cada clase, por lo que es necesario tratarlo como un problema de regresión.

En el caso de regresión, el modelo intenta predecir la posición exacta de cada piloto en la carrera, en lugar de clasificar su posición exacta. Para cada carrera, se ordenan los resultados de las predicciones en orden ascendente y se asigna la posición obtenida, es decir, aquel piloto con el valor más bajo se predice que ha terminado en primera posición y así sucesivamente con los otros 19 pilotos. A tal efecto, se utilizarán las versiones de los algoritmos explicados en la sección pero aplicados a la regresión. En la tabla 6.1 se muestran los modelos seleccionados y las abreviaturas utilizadas a lo largo de la memoria.

Modelo	Abreviatura
Decision Tree Regressor	DTR
Random Forest Regressor	RFR
XGBoost	XGB
Support Vector Regressor	SVR
Red Neuronal Artificial	RNA

Tabla 6.1: Modelos utilizados y sus abreviaturas

6.2 Métricas

Algunas de las métricas más utilizadas en los problemas de regresión y que sirven para evaluar el rendimiento de los modelos de predicción son las siguientes [45]:

- **Error Cuadrático Medio:** conocido como Mean Squared Error (MSE), mide la cantidad de error que hay entre dos conjuntos de datos, típicamente entre el conjunto real y el predicho. Se calcula como la suma de los errores al cuadrado dividido por el número de observaciones en el conjunto de datos. Cuanto menor sea el valor del MSE, mejor será la precisión del modelo.
- **Raíz del Error Cuadrático Medio:** métrica conocida como Root Mean Squared Error (RMSE), toma la raíz cuadrada del MSE. Esto significa que es una medida más intuitiva y fácil de interpretar, ya que los resultados se encuentran en la misma escala que los datos originales.
- $\boldsymbol{R^2}$**:** el coeficiente de determinación R^2 es una medida estadística que determina la proporción de varianza en la variable a predecir que se puede explicar por la variación de las variables independientes. En otras palabras, R^2 indica qué tan bien se ajusta la línea de regresión a los datos. El valor de R^2 varía entre 0 y 1, donde 0 indica que la línea de regresión no explica ninguna variación en la variable dependiente y 1 indica que la línea de regresión explica toda la variación en la variable dependiente. Por lo tanto, cuanto más cercano sea el valor de R^2 a 1, mejor se ajusta la línea de regresión a los datos.

A mayores, se considera una medida de correlación que explica la bondad para el caso de la predicción de posiciones, ya que son una variable ordinal. La *"correlación de Spearman"* [46]

es una medida de correlación no paramétrica que evalúa la relación monotónica entre dos variables ordinales. Si las posiciones predichas están altamente correlacionadas con las posiciones reales, entonces se puede decir que el modelo de predicción es preciso en términos de la relación monotónica entre las variables. En la tabla 6.2 se observan las métricas utilizadas y sobre que campos se calculan.

Posiciones predichas y verdaderas	Puntos predichos y verdaderos
Correlación de Spearman	R^2, *RMSE y MSE*

Tabla 6.2: Métricas utilizadas

6.3 Procesado

El primer paso es crear los conjuntos de entreno y test. Para el conjunto de entrenamiento contamos con los datos de las temporadas 2014 hasta 2021, tomando como inicio 2014 debido a la introducción del revolucionario reglamento de motorización comentado en la sección 5.2. Para evaluar el modelo se usa la información de la temporada 2022. Para esta tarea se ha utilizado principalmente la librería de Scikit-learn. A mayores, se ha tomado la decisión de no incluir las puntuaciones de las tres carreras *sprint* (con la mitad de vueltas y una asignación de puntos menor) realizadas a lo largo de la temporada 2022, es decir, solo se predicen las carreras de larga duración donde se sigue el reparto habitual de puntos a los pilotos.

El objetivo es crear modelos de predicción, por lo que los datos deben ser previamente preparados para ello. Dentro de las técnicas de preprocesado de datos en el ML, las más importantes son *Standard Scaler* y *One Hot Encoder*:

- ***Standard Scaler*:** normaliza las variables numéricas a la misma escala. Las variables numéricas de un conjunto de datos suelen tener diferentes escalas, algo problemático para algunos algoritmos. Al tipificar los datos, todas las variables tienen la misma escala y se vuelven comparables, algo que facilita el entrenamiento de los modelos.

- ***One Hot Encoder*:** transforma variables categóricas en numéricas. En el ML, los algoritmos trabajan únicamente con números, por lo que las variables categóricas deben transformarse para ser utilizadas en el entrenamiento. Se generan tantas variables binarias como posibles categorías tiene esa variable, donde un 1 indica que la variable tiene ese valor y un 0 lo contrario.

A continuación se crea un *pipeline*, que consiste en un conjunto de tuberías que permite combinar varias etapas de preprocesamiento y entrenamiento de modelos en un flujo unificado. Se usa para facilitar el proceso de construcción y permitir una experimentación opti-

mizada. La tabla 6.3 muestra las transformaciones aplicadas a las variables utilizadas en los modelos de predicción. Como aclaración, se ha optado por no incluir las variables de piloto y constructor ya que debido al cambio de reglamento de 2022, puede haber variación en los rendimientos de las temporadas utilizadas en el conjunto de entrenamiento, por lo que el modelo puede estar sesgado y realizar predicciones deficientes.

Standard Scaler	***One Hot Encoder***
pos_parrilla	temporada
dif_clasificacion	num_carrera
ult_3_carreras	circuito
ult_3_clasificaciones	
edad_piloto	
puntos_previos_piloto	
puntos_previos_constructor	

Tabla 6.3: Preprocesado de las características

Una vez creado el *pipeline* se realiza el entrenamiento con los diferentes modelos de regresión presentados en la tabla 6.1. Cada algoritmo tiene una serie de hiperparámetros, que se establecen antes de entrenar el modelo y no se aprenden de los datos. Estos parámetros determinan su comportamiento durante el entrenamiento y la capacidad de generalizar correctamente con datos nuevos. La selección de hiperparámetros es un proceso fundamental y que implica probar diferentes combinaciones de valores y evaluar el rendimiento del modelo en un conjunto de validación.

La técnica de *"k-fold cross validation"* [47] divide los datos de entrenamiento en k subconjuntos donde uno de los subconjuntos se utiliza como conjunto de validación y el resto de *k-1* subconjuntos se usan como conjunto de entrenamiento. El modelo se entrena k veces, cada una de ellas utilizando un subconjunto de validación diferente. El resultado final se obtiene promediando los resultados de las k repeticiones del entrenamiento.

Por lo tanto, se ha utilizado la técnica de *GridSearch_CV* con *k=5* subconjuntos para cada modelo de regresión propuesto. Con el mejor conjunto de hiperparámetros seleccionado, el siguiente paso es entrenar los modelos y obtener las predicciones.

Capítulo 7

Resultados

En este capítulo se comentarán los resultados obtenidos, utilizando para ello tablas con métricas y mostrando gráficamente las predicciones conseguidas. Se analizarán los resultados de ambos campeonatos, tanto el de pilotos como el de constructores. Al final de este capítulo se realizará una discusión de los resultados obtenidos.

7.1 Hiperparámetros óptimos

Tras el proceso de entrenamiento de los modelos, en la tabla 7.1 se puede observar el conjunto de hiperparámetros óptimo para cada algoritmo utilizado. Para el caso de la RNA, la estructura de las capas ocultas es [1024, 512, 256, 128, 64, 32, 16, 8] y como es un problema de regresión, cuenta con una única neurona en la capa de salida.

Modelo	*n_estimators*	*max_depth*	*learning_rate*	*kernel*	*C*
DTR	-	4	-	-	-
RFR	3000	4	-	-	-
XGB	50	2	0.1	-	-
SVR	-	-	-	linear	0.01
RNA	-	-	constant	-	-

Tabla 7.1: Hiperparámetros óptimos de los modelos

Tal y como se explicó en la sección 6.1, cada modelo intenta predecir las posiciones de los pilotos en cada carrera, por lo que se realiza la asignación de posiciones haciendo una ordenación de los valores predichos. El siguiente paso es convertir las posiciones predichas

de cada carrera en puntos. Dependiendo de la posición de llegada en la carrera, los primeros 10 pilotos consiguen una serie de puntos, tal y como se observa en la tabla 7.2.

Posición	**1º**	**2º**	**3º**	**4º**	**5º**	**6º**	**7º**	**8º**	**9º**	**10º**	**11º+**
Puntos	25	18	15	12	10	8	6	4	2	1	0

Tabla 7.2: Puntos repartidos según la posición de llegada

Una vez asignados los puntos de cada carrera según las posiciones predichas, se realiza una agregación de todas las carreras de la temporada 2022 para sumar los puntos y obtener la predicción de los campeonatos, tanto el de Pilotos (sección 7.2) como el de Constructores (sección 7.3). Por motivo del tamaño de las visualizaciones, sólo se comentarán y mostrarán los resultados del mejor y peor modelo para cada campeonato. No obstante, todas las tablas y gráficas de cada modelo serán mostradas en los apéndices de esta memoria, en concreto:

- Resultados DTR
- Resultados RFR
- Resultados XGB
- Resultados SVR
- Resultados RNA

7.2 Campeonato de Pilotos

Primero se realiza la agregación de los resultados predichos a lo largo de todas las carreras y se procede a la comparación de la predicción de la temporada frente a la realidad. Para el Campeonato de Pilotos, en la tabla 7.3 se pueden ver las métricas obtenidas por cada uno de los modelos.

Modelo	*Spearman*	R^2	*MSE*	*RMSE*
RNA	0.982	0.949	797.85	28.246
SVR	0.977	0.948	821.40	28.660
XGB	0.971	0.906	1483.60	38.518
RFR	0.953	0.890	1737.20	41.680
DTR	0.960	0.811	2895.10	54.636

Tabla 7.3: Resultados de los modelos sobre el Campeonato de Pilotos

Realizando una comparación de las métricas entre los distintos modelos, el mejor método sobre el conjunto de test resulta ser la RNA, ya que es el modelo que obtiene los valores más altos en la correlación de Spearman y en R^2, a la vez que consigue el valor de RMSE más bajo. En la figura 7.1 se observan los puntos y posiciones de cada piloto, tanto las predicciones como lo que ocurrió realmente en la temporada 2022, siendo la última columna la que indica el error cometido en las posiciones.

	Piloto	Puntos predichos	Puntos reales	Posición predicha	Posición real	Error: (Pred - Real)
0	max_verstappen	438	433	1	1	0
1	leclerc	378	291	2	2	0
2	perez	308	291	3	2	1
3	sainz	289	228	4	6	-2
4	russell	244	262	5	4	1
5	hamilton	201	233	6	5	1
6	norris	110	116	7	7	0
7	alonso	76	81	8	9	-1
8	ocon	49	89	9	8	1
9	bottas	39	47	10	10	0
10	ricciardo	31	34	11	12	-1
11	kevin_magnussen	20	21	12	14	-2
12	gasly	15	23	13	13	0
13	vettel	8	37	14	11	3
14	stroll	4	18	15	15	0
15	tsunoda	4	12	15	16	-1
16	mick_schumacher	3	12	17	16	1
17	albon	2	4	18	19	-1
18	zhou	2	6	18	18	0
19	latifi	0	2	20	20	0

Figura 7.1: Mejor predicción obtenida para Pilotos

Se observa que predice con bastante precisión los puntos de cada piloto y por lo tanto, tiene un bajo error asignando las posiciones finales en el campeonato, de ahí que en la columna

de *"Error: (Pred - Real)"*, la inmensa mayoría de valores sean 0 o ±1, es decir, prediciendo la posición real o cometiendo un error de una posición. Teniendo en cuenta un margen de una posición arriba o abajo, el modelo predice correctamente el 85% de las posiciones finales.

Gráficamente, la figura 7.2 muestra la comparativa de puntos predichos y reales para cada piloto, usando para ello la función *"regplot"* de Seaborn. La línea de color naranja sirve para visualizar la relación entre dos variables continuas, mostrando la línea de regresión y su intervalo de confianza al 95%, que indica la incertidumbre de las estimaciones del modelo, siendo el intervalo más amplio en aquellas regiones donde hay menos puntos, lo que supone una mayor incertidumbre en esas zonas. Por otro lado, la línea en color azul se trata de una diagonal que indica lo que sería el *"ajuste perfecto"* y sirve para evaluar rápidamente la precisión del modelo. Si el punto está situado sobre la recta, la predicción es idéntica a la realidad. Si los puntos estimados y los reales se encuentran cercanos a esta línea, esto indica que las predicciones son precisas. Por último, si los puntos están muy distanciados de la diagonal, el modelo no está realizando unas buenas predicciones.

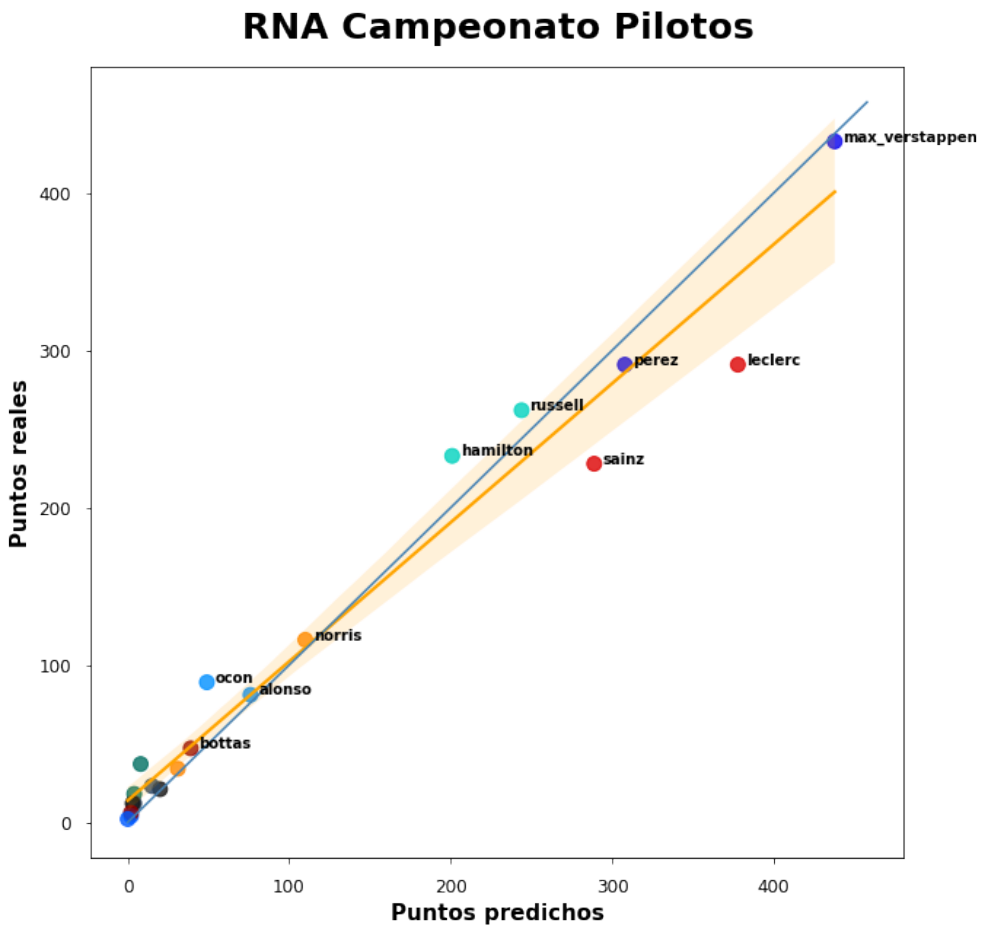


Figura 7.2: Gráfica de puntos del mejor modelo para Pilotos

En la figura 7.2 se observa que la mayoría de los pilotos se encuentran dentro del intervalo de confianza, por lo que el modelo de la RNA ajusta bien a los datos y sus predicciones tienen poca varianza. Destaca el caso de los pilotos de Ferrari (Sainz y Leclerc), que se encuentran por debajo de la diagonal, indicando que el modelo les asignó una cantidad de puntos mayor de lo que realmente pudieron lograr a lo largo de las carreras, es decir, rindieron por debajo de las expectativas del modelo. Sucede lo contrario con los pilotos de Mercedes (Hamilton y Russell), situados por encima de la línea azul, obteniendo mejores resultados de los estimados por las variables.

Para comparar estos resultados con otro modelo con peor desempeño, observaremos los mismos tipos de gráficas pero usando las del DTR, el peor algoritmo según las métricas.

	Piloto	Puntos predichos	Puntos reales	Posición predicha	Posición real	Error: (Pred - Real)
0	max_verstappen	426	433	1	1	0
1	sainz	418	228	2	6	-4
2	leclerc	408	291	3	2	1
3	perez	315	291	4	2	2
4	hamilton	224	233	5	5	0
5	russell	202	262	6	4	2
6	norris	148	116	7	7	0
7	alonso	132	81	8	9	-1
8	ocon	68	89	9	8	1
9	bottas	55	47	10	10	0
10	ricciardo	49	34	11	12	-1
11	gasly	42	23	12	13	-1
12	kevin_magnussen	32	21	13	14	-1
13	mick_schumacher	12	12	14	16	-2
14	vettel	11	37	15	11	4
15	stroll	10	18	16	15	1
16	tsunoda	9	12	17	16	1
17	albon	8	4	18	19	-1
18	zhou	4	6	19	18	1
19	latifi	3	2	20	20	0

Figura 7.3: Peor predicción obtenida para Pilotos

Comparando los resultados de la tabla del DTR (figura 7.3) con la tabla que mostraba los resultados de la RNA (figura 7.1), los errores cometidos en la predicción de puntos y en la consecuente posición en el campeonato son mayores, obteniendo un 75% de acierto teniendo en cuenta el margen de un fallo en la posición, resultado inferior comparado con el 85% anterior.

Gráficamente, en la figura 7.4 también se puede apreciar la diferencia de las predicciones. Si la comparamos con la figura de la RNA (figura 7.2), la línea naranja que indica la relación entre los puntos reales y predichos se aleja bastante de la línea azul, que indica la regresión ideal. El caso que más destaca es la coordenada de Sainz, muy alejada de la realidad debido a que el modelo le asignó demasiados puntos a lo largo del campeonato.

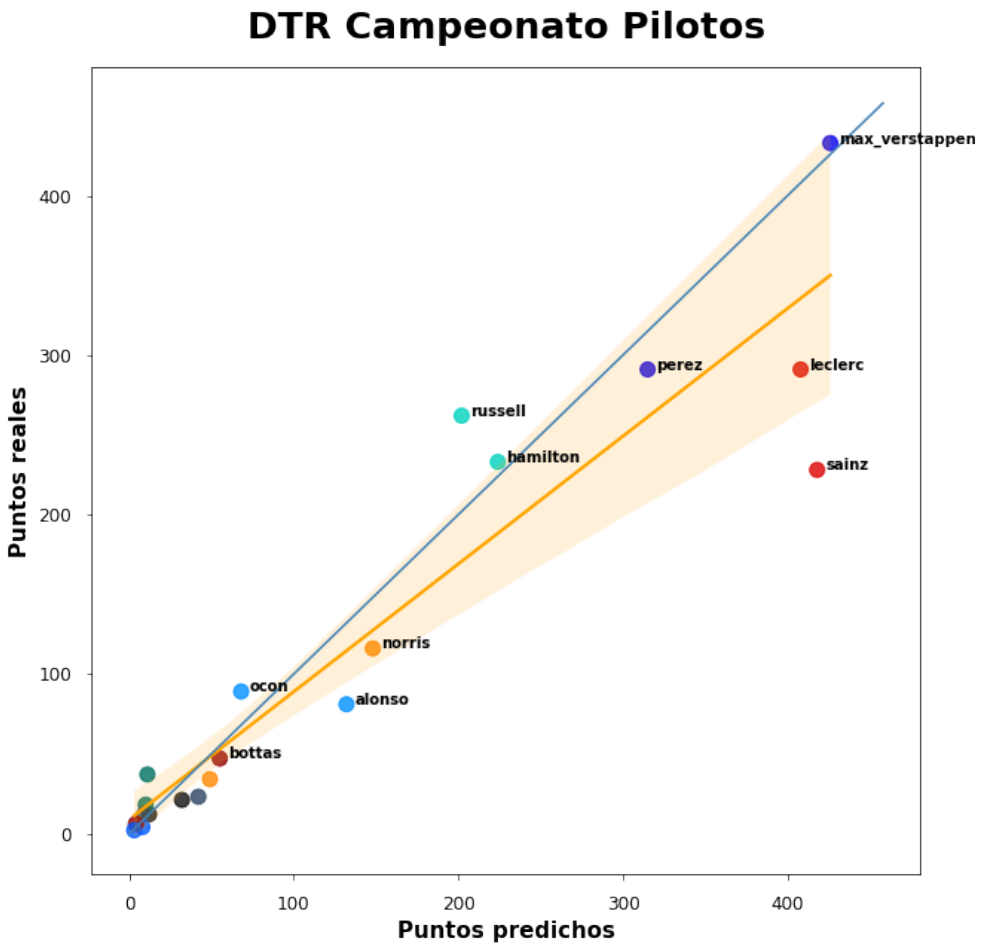


Figura 7.4: Gráfica de puntos del peor modelo para Pilotos

7.3 Campeonato de Constructores

Se realiza el mismo esquema pero aplicado a los equipos. Tal y como se explicó en la sección 2.1, la puntuación de los pilotos del equipo se suma para formar la puntuación total del constructor. Las métricas obtenidas para el Campeonato de Constructores se pueden observar en la tabla 7.4.

Modelo	*Spearman*	R^2	*MSE*	*RMSE*
SVR	0.903	0.961	2321.4	48.181
RNA	0.903	0.951	2935.2	54.177
XGB	0.891	0.909	5419.2	73.615
RFR	0.891	0.895	6251.0	79.063
DTR	0.915	0.826	10400.2	101.981

Tabla 7.4: Resultados de los modelos sobre el Campeonato de Constructores

Los modelos de SVR y RNA obtienen el mismo valor para la correlación de Spearman ya que las posiciones predichas para los equipos son idénticas para ambos modelos, no obstante, es en la predicción de puntos conseguidos donde SVR consigue un resultado ligeramente más ajustado a la realidad (un valor de RMSE más bajo). En la figura 7.5 se pueden observar las mejores predicciones de las posiciones y puntos de los equipos para la temporada 2022.

	Constructor	Puntos predichos	Puntos reales	Posición predicha	Posición real	Error: (Pred - Real)
0	red_bull	743	724	1	1	0
1	ferrari	641	519	2	2	0
2	mercedes	426	495	3	3	0
3	mclaren	146	150	4	5	-1
4	alpine	131	170	5	4	1
5	alfa_romeo	43	53	6	7	-1
6	haas	36	33	7	9	-2
7	alpha_tauri	30	35	8	8	0
8	aston_martin	16	55	9	6	3
9	williams	10	6	10	10	0

Figura 7.5: Mejor predicción obtenida para Constructores

Si se tiene en cuenta un margen de error de una posición en la predicción, el acierto del modelo aumenta hasta el 80%. En la figura 7.6 se observa la gráfica de puntos predichos y reales de los constructores.

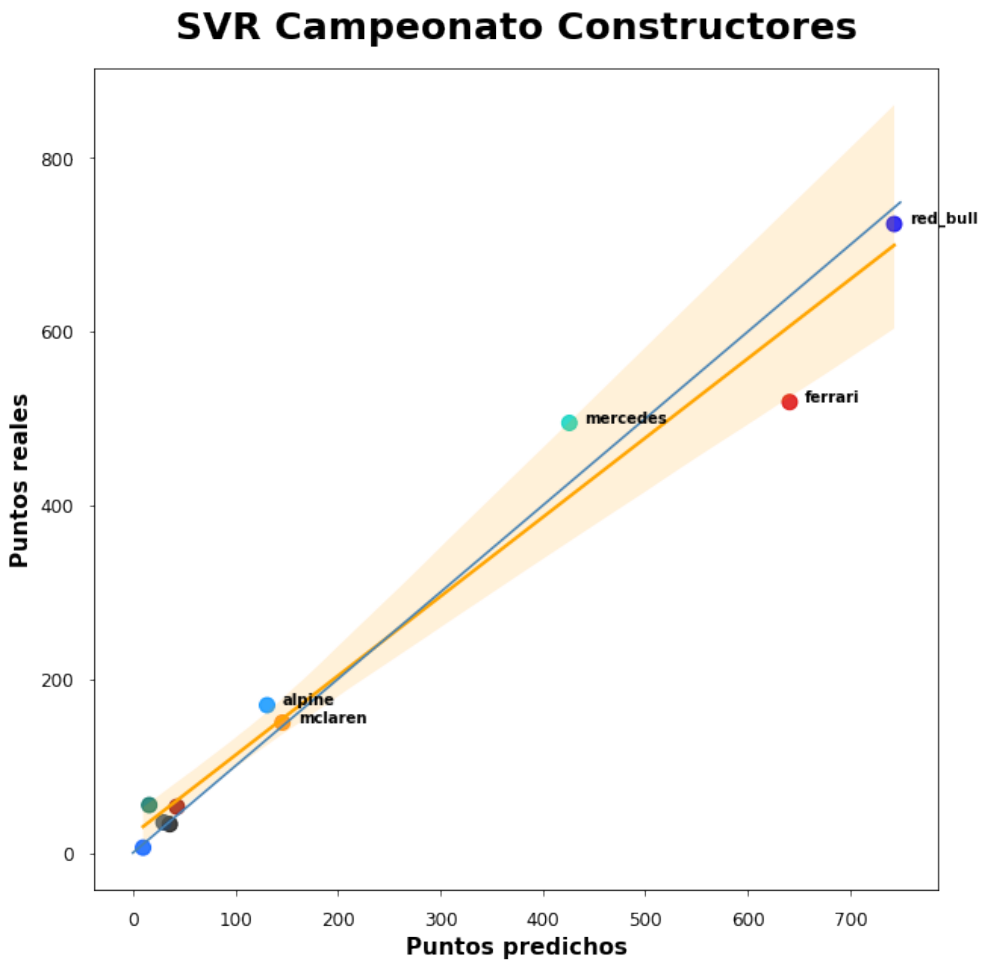


Figura 7.6: Gráfica de puntos del mejor modelo para Constructores

Todos los equipos se encuentran bastante cerca de la línea naranja que indica la regresión, que a su vez es muy parecida a la línea azul que se trata de la línea divisoria ideal. Como curiosidad, se puede ver como Ferrari no estuvo de acorde a las expectativas y como Mercedes obtuvo un mejor rendimiento en puntos de lo esperado.

Al igual que en la sección 7.2, se compara el mejor modelo con el que peor rendimiento obtuvo. Según las métricas de la tabla 7.4, el método con peor rendimiento para el caso de constructores es de nuevo el DTR. Las predicciones obtenidas por el modelo se pueden ver en la figura 7.7.

	Constructor	Puntos predichos	Puntos reales	Posición predicha	Posición real	Error: (Pred - Real)
0	ferrari	826	519	1	2	-1
1	red_bull	741	724	2	1	1
2	mercedes	426	495	3	3	0
3	alpine	200	170	4	4	0
4	mclaren	197	150	5	5	0
5	alfa_romeo	59	53	6	7	-1
6	alpha_tauri	51	35	7	8	-1
7	haas	44	33	8	9	-1
8	aston_martin	21	55	9	6	3
9	williams	11	6	10	10	0

Figura 7.7: Peor predicción obtenida para Constructores

El valor tan elevado de *RMSE = 101.981* se debe en gran medida al error cometido sobre el equipo Ferrari, asignándole una cantidad de puntos desorbitada respecto a lo que sucedió realmente en la temporada. En la figura 7.8 se observa el rendimiento del modelo.

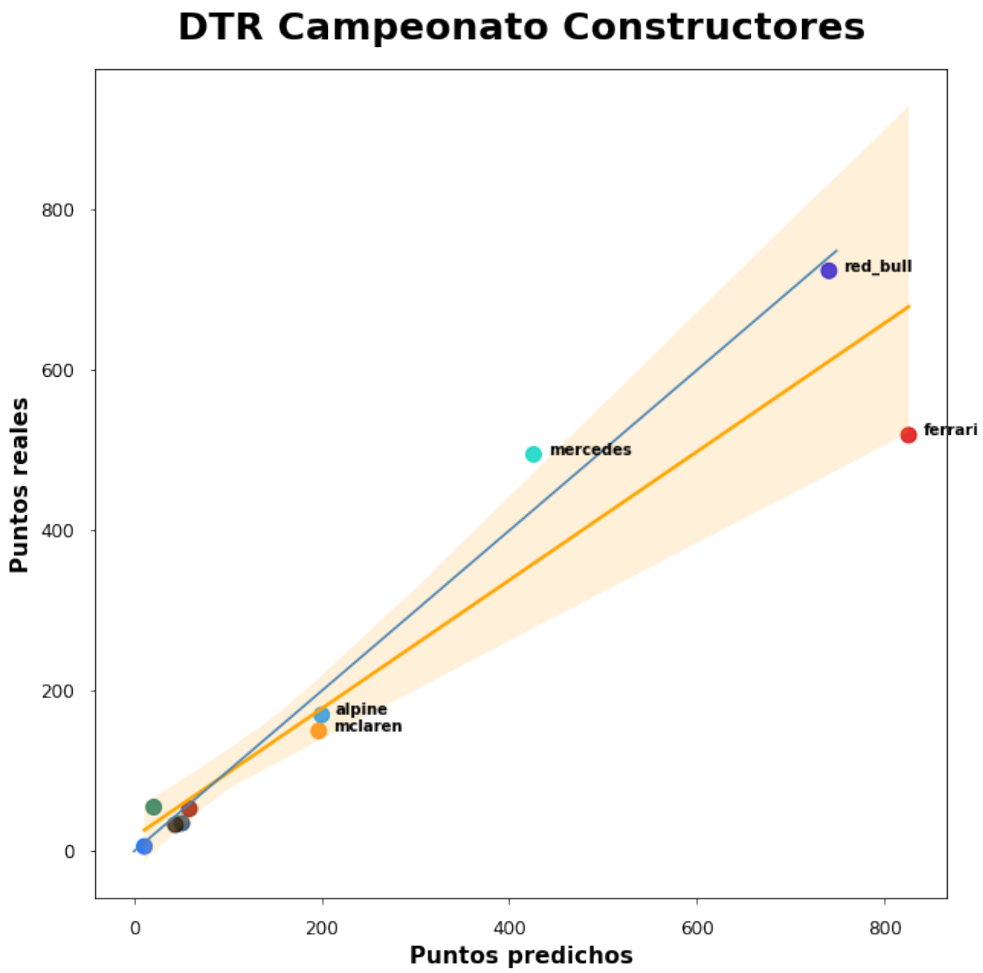


Figura 7.8: Gráfica de puntos del peor modelo para Constructores

Al contrario de lo que sucede en la figura 7.6, en la gráfica del DTR (figura 7.8) la línea naranja difiere de la línea azul y los puntos se encuentran más alejados, estando en el límite de la región del intervalo de confianza del 95%.

7.4 Discusión de los resultados

Los resultados obtenidos en ambos campeonatos han sido satisfactorios. En los dos casos, los modelos con mejor rendimiento en las predicciones han sido la RNA y el SVR. Por otro lado, los métodos basados en árboles (XGB, RFR y DTR) han obtenido peores predicciones, destacando el DTR como el peor modelo de toda la comparativa, con unos valores de RMSE muy altos y con indicadores de Spearman y R^2 pobres comparado con el resto de modelos.

El mejor modelo para el Campeonato de Pilotos (sección 7.2) fue la RNA, prediciendo las posiciones finales y los puntos obtenidos con un gran grado de precisión con respecto a la

realidad. En el caso del Campeonato de Constructores (sección 7.3) la RNA también obtiene un gran resultado, muy cercano al del SVR, pero como los puntos obtenidos por el equipo son la suma de los dos pilotos, nos interesa más predecir correctamente el Campeonato de Pilotos, y por lo tanto, el mejor modelo para ello es la RNA. Pese a ser un modelo de caja negra (explicado en la subsección 2.2.5), hay un método para obtener la importancia de las variables después de entrenar el modelo. Se consigue mediante la técnica de *"permutación de importancia de características"* [48, 49], que se basa en medir la precisión del modelo antes y después de realizar permutaciones aleatorias en las variables del conjunto de datos. Si se permuta un valor de la variable y esto causa una disminución importante en la precisión del modelo, se considera que dicha característica del conjunto es importante para el rendimiento del modelo. Los pasos a seguir son:

1. Entrenar el modelo con las características originales y evaluar la precisión usando un conjunto de datos de prueba.
2. Seleccionar una característica y permutar aleatoriamente los valores en el conjunto de prueba.
3. Volver a medir la precisión del modelo en el conjunto de pruebas permutado.
4. Este proceso se repite para cada una de las características del conjunto de datos.
5. Se calcula la relevancia de cada variable como la diferencia de la precisión del modelo original y el permutado, normalizando el resultado por la desviación estándar de las importancias de las características. A mayores se divide el resultado entre el total para así que todo sume 1 y poder compararlo con el resto de modelos.

Para utilizar esta técnica se puede usar la función *"permutation_importance"* de Scikit-learn. Los resultados obtenidos de la importancia de las variables de la RNA se pueden observar en la figura 7.9.

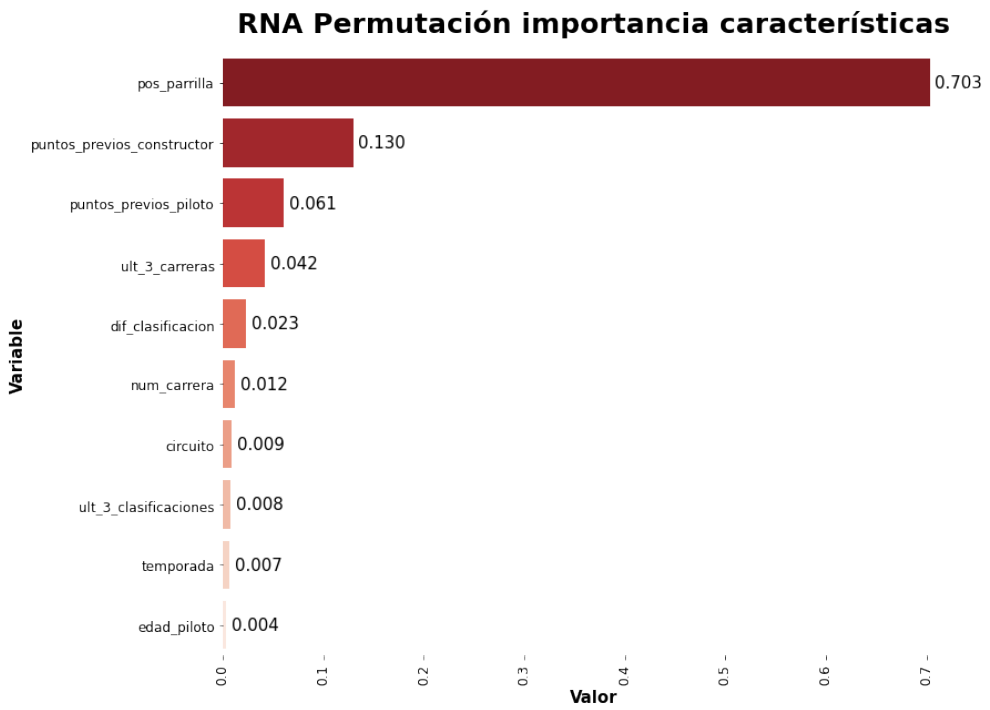
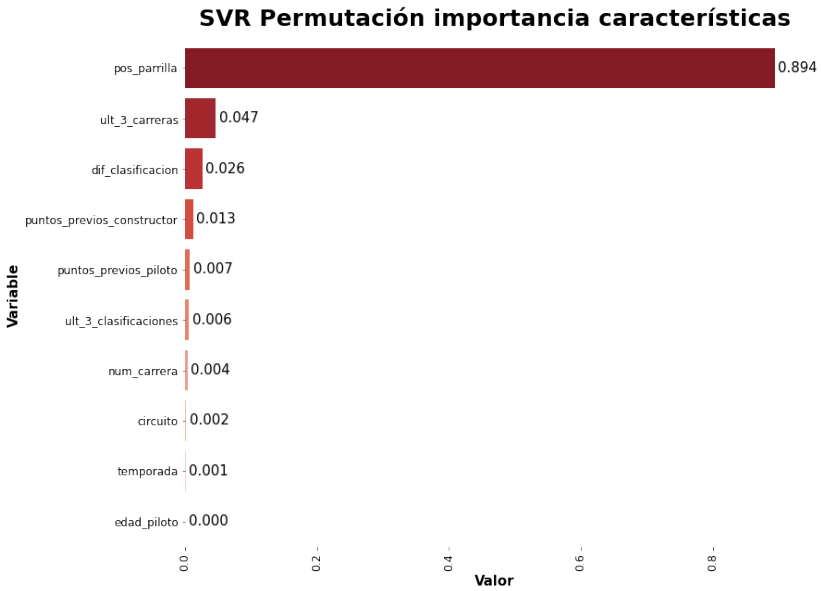


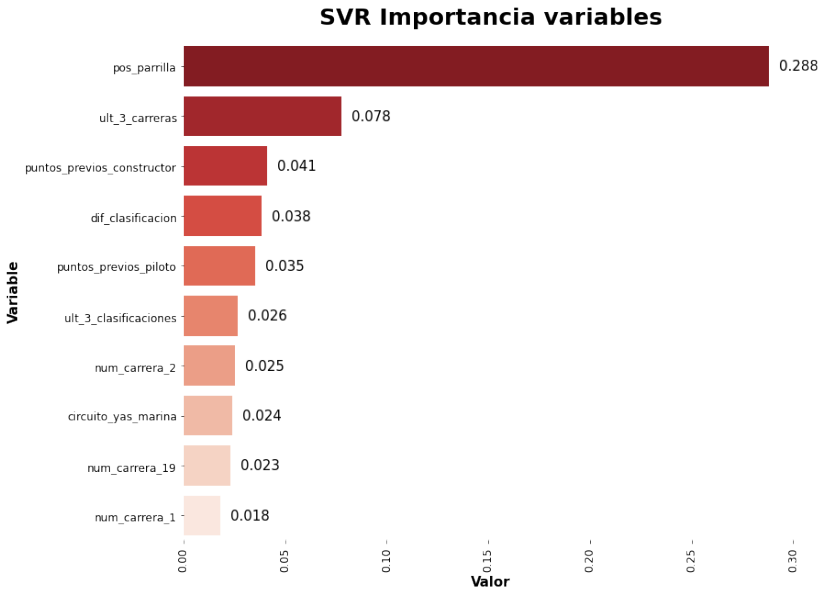
Figura 7.9: RNA: importancia de las variables

La variable más importante y con gran diferencia al resto es *"pos_parrilla"*, con un 70.3% de la relevancia total del conjunto de datos. Esto concuerda con lo comentado en la subsección 5.2.1, donde la característica más correlacionada con la variable a predecir era la posición de parrilla. Luego están las variables que reflejan los puntos previos del equipo y del piloto, con una importancia del 13% y 6.1% respectivamente. En cuarto lugar está *"ult_3_carreras"* con un 4.2% de relevancia, que refleja la diferencia media de tiempos de finalización de las últimas tres carreras con respecto al vencedor.

El segundo mejor modelo de la comparativa es el SVR, con resultados muy buenos pero ligeramente inferiores a la RNA. También se puede obtener la relevancia de las variables usando la técnica comentada anteriormente. A mayores, este modelo tiene un atributo de coeficientes, que se debe tomar el valor absoluto de cada atributo y normalizar los coeficientes para que entre todos sumen 1 y así obtener la importancia relativa y poder comparar con el resto de modelos de ML utilizados. En la figura 7.10 se comparan ambos métodos para obtener la relevancia de las variables sobre el SVR.



(a) Usando la permutación



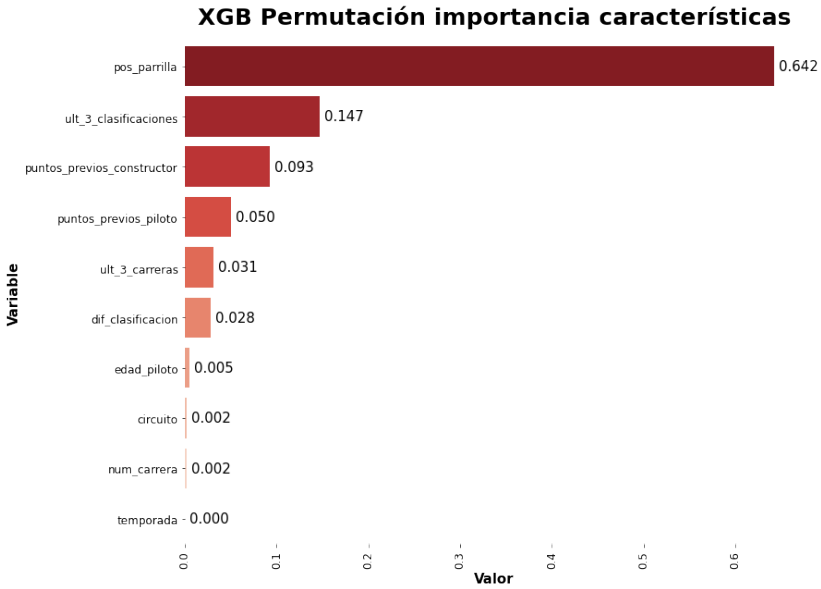
(b) Usando coeficientes

Figura 7.10: SVR: importancia de las variables

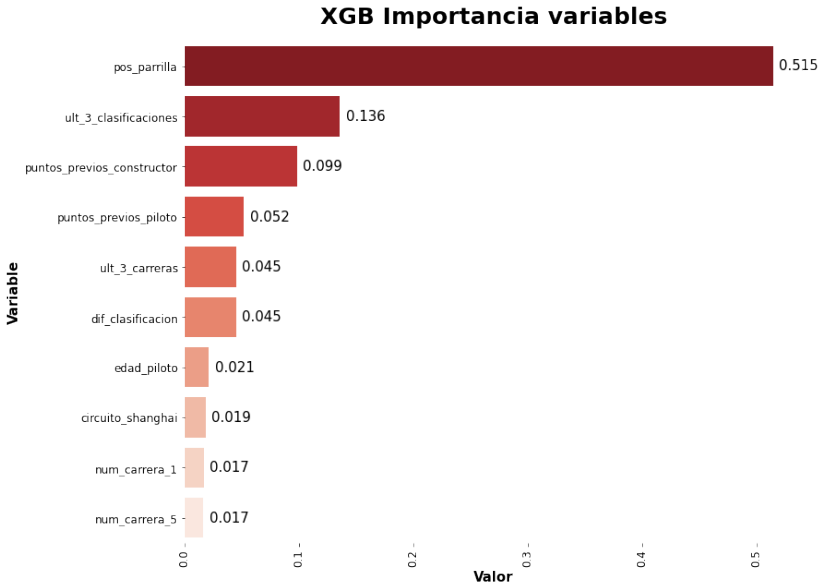
Con ámbos métodos se obtiene que las dos variables más relevantes para este modelo son *"pos_parrilla"* y *"ult_3_carreras"*. Los coeficientes de importancia de las variables varían ya que se consiguen con distintas técnicas.

En los algoritmos basados en árboles, la importancia de las variables se puede medir directamente a partir del modelo del árbol entrenado. Dentro de los métodos de árboles utilizados, el que mejor rendimiento obtuvo fue el XGB. El método para obtener las importancias es acce-

der al mejor estimador del modelo y extraer los valores de las características. La otra técnica es la ya comentada y utilizada en las anteriores gráficas. Los resultados se observan en la figura 7.11.



(a) Usando la permutación



(b) Extrayendo la importancia del modelo entrenado

Figura 7.11: XGB: importancia de las variables

Los dos métodos utilizados sobre XGB devuelven el mismo *ranking* de importancia de las variables, siendo los coeficientes muy parecidos entre sí. Sorprende que la variable de las tres

últimas clasificaciones tenga una importancia tan elevada y que su variante de últimas tres carreras esté situada tan abajo.

Con respecto a los modelos en sí, cada modelo tiene sus fortalezas y debilidades. Por ejemplo, el mejor modelo es la RNA, caracterizada por ser capaz de aprender patrones complejos y no lineales de los datos, por ser flexible en la arquitectura y por tener muchos hiperparámetros configurables. La contra de este modelo es la cantidad de cómputo necesario y la dificultad de entender su funcionamiento. Por otro lado están los modelos basados en árboles, que son más fáciles de interpretar pero que no tienen tanto poder de predicción. No obstante, se observa que los resultados que proporcionan XGB y RFR son mejores que los del DTR, ya que estos modelos son de tipo *ensemble* y son más complejos que un simple árbol, dando por lo tanto mejores resultados.

Como conclusión, hay situaciones donde un modelo puede ser una mejor opción que otro, dependiendo en gran medida del conjunto de datos a utilizar y del problema específico abordado. En este caso, el mejor modelo fue la RNA y la variable más relevante fue la posición de parrilla en el inicio de la carrera, seguido por variables que denotan el rendimiento de las carreras más recientes.

Capítulo 8

Conclusiones

Este último capítulo tratará de las conclusiones obtenidas tras la realización del proyecto, así como los conocimientos adquiridos relacionados con el grado universitario. Por último, se propondrán posibles mejoras y líneas futuras del proyecto.

8.1 Conclusiones sobre el proyecto

Este proyecto se ha centrado en investigar la capacidad de predicción que tienen diversos modelos de SL sobre los resultados de una temporada de F1 aplicando regresión. Se ha llevado a cabo un estudio de las diferentes técnicas disponibles en el campo del ML, además de un análisis sobre la predicción de resultados en diferentes deportes para analizar técnicas que pudieran mejorar los resultados.

Para la carga del conjunto de datos se utilizó la información considerada más relevante para el caso, dejando fuera del modelo datos como los incidentes de carrera, tiempos de cada vuelta de la carrera o duración de los cambios de rueda en los *pit stop*. Toda la información fue procesada previamente al entrenamiento de los modelos y a mayores se crearon nuevas variables que ayudaron a mejorar la precisión de las predicciones. Para mejorar la comprensión de los datos y del deporte en cuestión, se crearon unas visualizaciones que muestran aspectos destacables como la correlación de las variables y las tendencias en las últimas temporadas. Mediante el uso de *pipelines* y de técnicas como la validación cruzada, se ajustaron los hiperparámetros de los distintos modelos de una forma eficiente.

Aplicando regresión en cada una de las carreras disputadas en la temporada, el proceso seguido fue asignar puntos según las posiciones predichas y realizar la agregación de los puntos al final de la temporada para ambos campeonatos, tanto el de pilotos como el de constructores. En el Campeonato de Pilotos (sección 7.2), los modelos que arrojaron unos resultados más satisfactorios con respecto a la realidad fueron la RNA y el SVR, obteniendo valores muy altos para la correlación de Spearman (0.982) y para el R^2 (0.949), a la vez que el RMSE logrado

sobre la predicción de puntos era relativamente bajo (28.246). El mejor modelo fue la RNA, siendo capaz de predecir correctamente el 85% de las posiciones finales teniendo en cuenta un margen de error de una posición, además de que logró predecir correctamente los tres primeros pilotos del campeonato. Por otro lado, los modelos basados en árboles (XGB, RFR y DTR) consiguieron predicciones más discretas, con peores métricas, siendo el DTR el peor de todos los modelos debido a su simplicidad frente a los otros métodos empleados. Para el Campeonato de Constructores (sección 7.3) las conclusiones obtenidas son las mismas.

Por último, observando la importancia de las características sobre los distintos modelos y la matriz de correlaciones de las variables, se llegó a la conclusión de que un factor muy importante para determinar la posición de llegada en la carrera es la posición de salida obtenida a lo largo de la clasificación, seguida de cerca por variables que indican los resultados obtenidos en las últimas carreras, que indican la dominancia de los pilotos y los constructores en las clasificaciones y en las carreras.

En resumen, pese a que se ha investigado en profundidad sobre la predicción de resultados, obteniendo resultados satisfactorios muy parecidos a la realidad, la verdad es que predecir con completa exactitud los resultados de una temporada de F1 es una tarea extremadamente difícil. Es un deporte en el que intervienen muchos factores impredecibles, siendo esta incertidumbre lo que hace que esta competición sea tan emocionante, tanto para los pilotos como para los fans de la F1. Con todo esto se ha desarrollado una metodología que permite evaluar de forma objetiva el rendimiento de los pilotos y equipos a lo largo de una temporada, comparando los resultados reales con las expectativas generadas por los algoritmos de predicción.

8.2 Conocimientos adquiridos

Respecto a las competencias de la titulación, he podido afianzar los conocimientos aprendidos a lo largo de las diferentes asignaturas del grado, en concreto las siguientes:

- Aprendizaje Automático
- Bases de Datos
- Probabilidad y Estadística

8.3 Trabajo futuro

Pese a que los resultados obtenidos son más que satisfactorios y cumplen con los objetivos planteados al inicio del proyecto, a lo largo del proceso han surgido posibles ideas que pueden suponer mejoras. Por lo tanto, aquí se presentan el posible trabajo futuro a realizar teniendo como base este proyecto:

- **Información de la climatología:** sería interesante añadir información sobre el clima de las distintas carreras, para comprobar si el rendimiento de los pilotos varía en condiciones adversas.
- **Inclusión de más variables:** tener en cuenta información como los tiempos individuales de cada vuelta o la probabilidad de coche de seguridad en cada circuito pueden alterar las predicciones de los modelos. También puede ser relevante usar información de los entrenamientos libres para analizar cómo los equipos mejoran el rendimiento del coche antes de la clasificación y de la carrera. Igualmente, la información sobre la degradación de los neumáticos también es clave para el desarrollo de una carrera.
- **Fuentes de datos externas:** *F1 Insights powered by AWS* [50] es una plataforma de análisis desarrollada por Amazon en colaboración con la F1. Combina información histórica con datos en tiempo real, proporcionando información sobre el rendimiento en curvas, el frenado, ritmo de clasificación, etc. Todo esto puede ser utilizado para enriquecer los modelos de predicción.
- **Probar más combinaciones de hiperparámetros:** añadir más posibilidades en los diferentes hiperparámetros de los modelos para obtener mejores predicciones. Esto implica una mayor carga de computación y de tiempos de espera. También sería recomendable probar más configuraciones para la RNA.
- **Creación de una aplicación web:** para poder ir prediciendo los resultados de las carreras a la vez que la temporada avanza y visualizar las predicciones.

Apéndices

Apéndice A

Resultados DTR

	Piloto	Puntos predichos	Puntos reales	Posición predicha	Posición real	Error: (Pred - Real)
0	max_verstappen	426	433	1	1	0
1	sainz	418	228	2	6	-4
2	leclerc	408	291	3	2	1
3	perez	315	291	4	2	2
4	hamilton	224	233	5	5	0
5	russell	202	262	6	4	2
6	norris	148	116	7	7	0
7	alonso	132	81	8	9	-1
8	ocon	68	89	9	8	1
9	bottas	55	47	10	10	0
10	ricciardo	49	34	11	12	-1
11	gasly	42	23	12	13	-1
12	kevin_magnussen	32	21	13	14	-1
13	mick_schumacher	12	12	14	16	-2
14	vettel	11	37	15	11	4
15	stroll	10	18	16	15	1
16	tsunoda	9	12	17	16	1
17	albon	8	4	18	19	-1
18	zhou	4	6	19	18	1
19	latifi	3	2	20	20	0

(a) Campeonato de Pilotos

	Constructor	Puntos predichos	Puntos reales	Posición predicha	Posición real	Error: (Pred - Real)
0	ferrari	826	519	1	2	-1
1	red_bull	741	724	2	1	1
2	mercedes	426	495	3	3	0
3	alpine	200	170	4	4	0
4	mclaren	197	150	5	5	0
5	alfa_romeo	59	53	6	7	-1
6	alpha_tauri	51	35	7	8	-1
7	haas	44	33	8	9	-1
8	aston_martin	21	55	9	6	3
9	williams	11	6	10	10	0

(b) Campeonato de Constructores

Figura A.1: DTR: tablas de predicciones

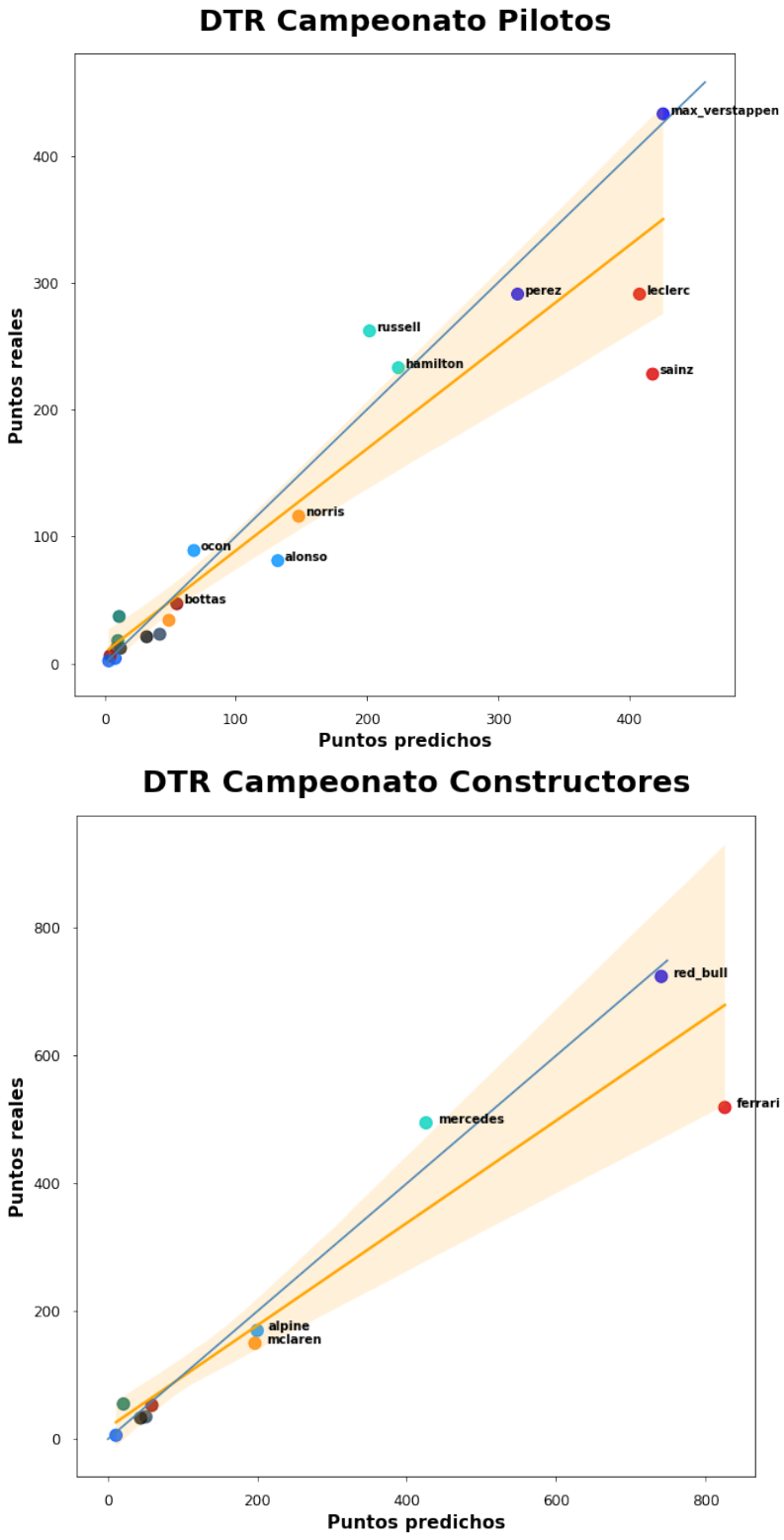


Figura A.2: DTR: gráficas de puntos

Apéndice B

Resultados RFR

	Piloto	Puntos predichos	Puntos reales	Posición predicha	Posición real	Error: (Pred - Real)
0	leclerc	402	291	1	2	-1
1	max_verstappen	369	433	2	1	1
2	sainz	341	228	3	6	-3
3	perez	294	291	4	2	2
4	russell	219	262	5	4	1
5	hamilton	208	233	6	5	1
6	norris	116	116	7	7	0
7	alonso	80	81	8	9	-1
8	ocon	46	89	9	8	1
9	bottas	43	47	10	10	0
10	ricciardo	28	34	11	12	-1
11	kevin_magnussen	25	21	12	14	-2
12	gasly	17	23	13	13	0
13	mick_schumacher	8	12	14	16	-2
14	tsunoda	8	12	14	16	-2
15	vettel	7	37	16	11	5
16	stroll	5	18	17	15	2
17	albon	4	4	18	19	-1
18	zhou	2	6	19	18	1
19	latifi	0	2	20	20	0

(a) Campeonato de Pilotos

	Constructor	Puntos predichos	Puntos reales	Posición predicha	Posición real	Error: (Pred - Real)
0	ferrari	743	519	1	2	-1
1	red_bull	663	724	2	1	1
2	mercedes	427	495	3	3	0
3	mclaren	144	150	4	5	-1
4	alpine	126	170	5	4	1
5	alfa_romeo	45	53	6	7	-1
6	haas	33	33	7	9	-2
7	alpha_tauri	25	35	8	8	0
8	aston_martin	12	55	9	6	3
9	williams	4	6	10	10	0

(b) Campeonato de Constructores

Figura B.1: RFR: tablas de predicciones

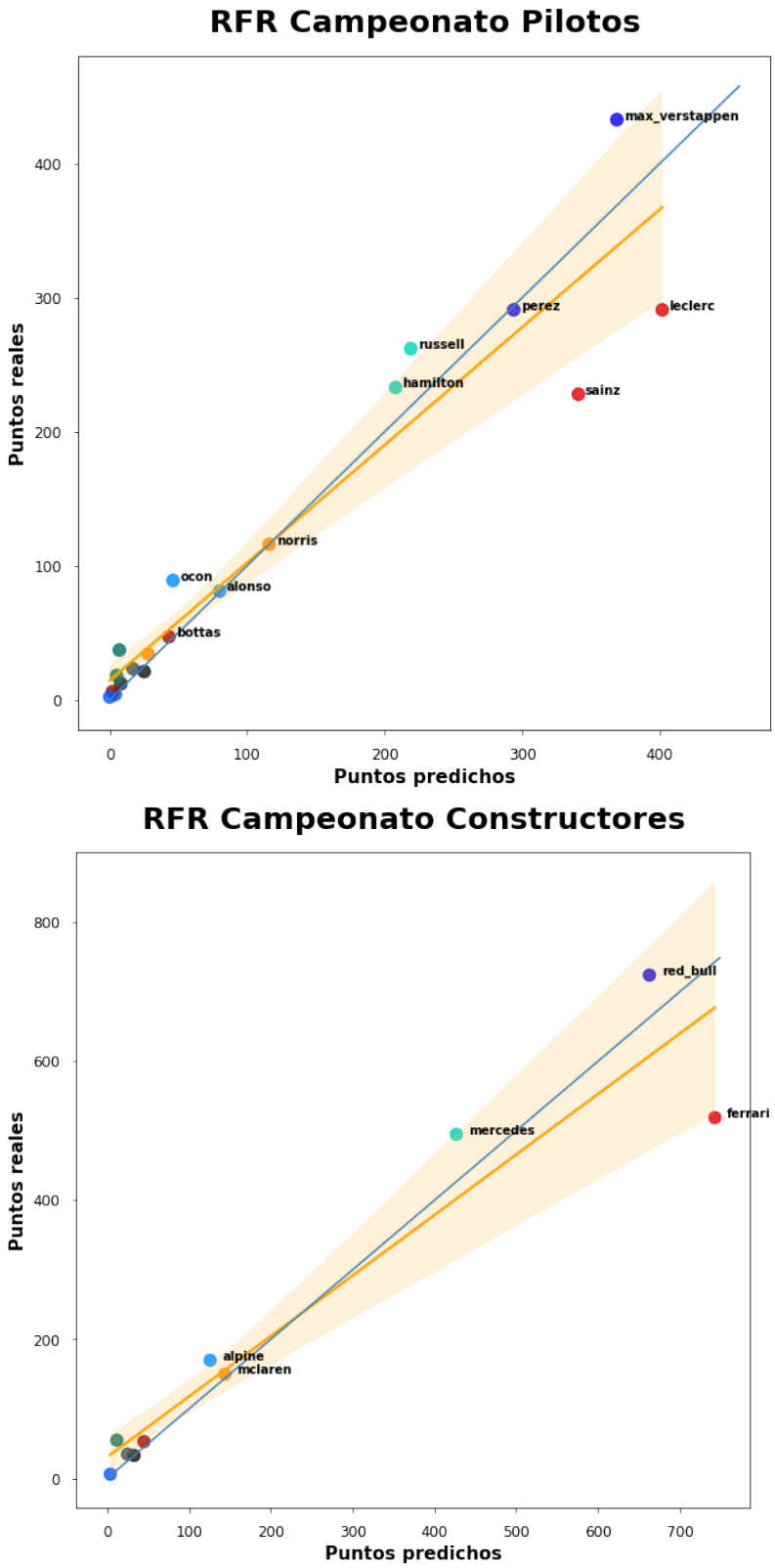


Figura B.2: RFR: gráficas de puntos

Apéndice C

Resultados XGB

	Piloto	Puntos predichos	Puntos reales	Posición predicha	Posición real	Error: (Pred - Real)
0	max_verstappen	401	433	1	1	0
1	leclerc	397	291	2	2	0
2	sainz	329	228	3	6	-3
3	perez	290	291	4	2	2
4	hamilton	208	233	5	5	0
5	russell	201	262	6	4	2
6	norris	112	116	7	7	0
7	alonso	94	81	8	9	-1
8	ocon	52	89	9	8	1
9	bottas	41	47	10	10	0
10	ricciardo	35	34	11	12	-1
11	kevin_magnussen	23	21	12	14	-2
12	gasly	15	23	13	13	0
13	vettel	7	37	14	11	3
14	tsunoda	6	12	15	16	-1
15	mick_schumacher	5	12	16	16	0
16	stroll	4	18	17	15	2
17	albon	2	4	18	19	-1
18	zhou	2	6	18	18	0
19	latifi	0	2	20	20	0

(a) Campeonato de Pilotos

	Constructor	Puntos predichos	Puntos reales	Posición predicha	Posición real	Error: (Pred - Real)
0	ferrari	726	519	1	2	-1
1	red_bull	691	724	2	1	1
2	mercedes	409	495	3	3	0
3	mclaren	147	150	4	5	-1
4	alpine	146	170	5	4	1
5	alfa_romeo	43	53	6	7	-1
6	haas	28	33	7	9	-2
7	alpha_tauri	21	35	8	8	0
8	aston_martin	11	55	9	6	3
9	williams	2	6	10	10	0

(b) Campeonato de Constructores

Figura C.1: XGB: tablas de predicciones

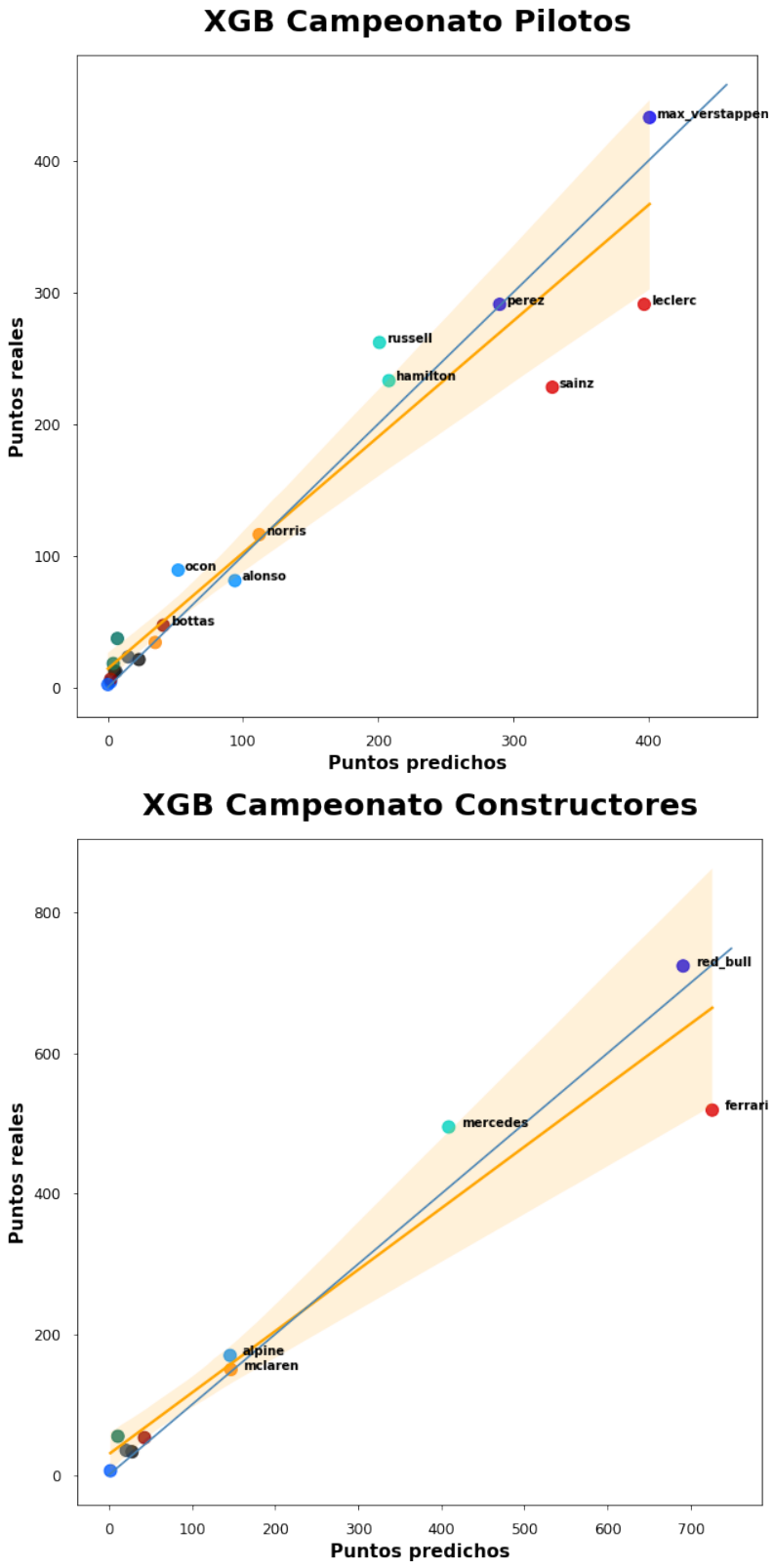


Figura C.2: XGB: gráficas de puntos

Apéndice D

Resultados SVR

	Piloto	Puntos predichos	Puntos reales	Posición predicha	Posición real	Error: (Pred - Real)
0	max_verstappen	410	433	1	1	0
1	leclerc	368	291	2	2	0
2	perez	333	291	3	2	1
3	sainz	273	228	4	6	-2
4	hamilton	215	233	5	5	0
5	russell	211	262	6	4	2
6	norris	111	116	7	7	0
7	alonso	88	81	8	9	-1
8	ocon	43	89	9	8	1
9	bottas	40	47	10	10	0
10	ricciardo	35	34	11	12	-1
11	kevin_magnussen	30	21	12	14	-2
12	gasly	21	23	13	13	0
13	vettel	10	37	14	11	3
14	tsunoda	9	12	15	16	-1
15	stroll	6	18	16	15	1
16	mick_schumacher	6	12	16	16	0
17	albon	4	4	18	19	-1
18	zhou	3	6	19	18	1
19	latifi	0	2	20	20	0

(a) Campeonato de Pilotos

	Constructor	Puntos predichos	Puntos reales	Posición predicha	Posición real	Error: (Pred - Real)
0	red_bull	743	724	1	1	0
1	ferrari	641	519	2	2	0
2	mercedes	426	495	3	3	0
3	mclaren	146	150	4	5	-1
4	alpine	131	170	5	4	1
5	alfa_romeo	43	53	6	7	-1
6	haas	36	33	7	9	-2
7	alpha_tauri	30	35	8	8	0
8	aston_martin	16	55	9	6	3
9	williams	10	6	10	10	0

(b) Campeonato de Constructores

Figura D.1: SVR: tablas de predicciones

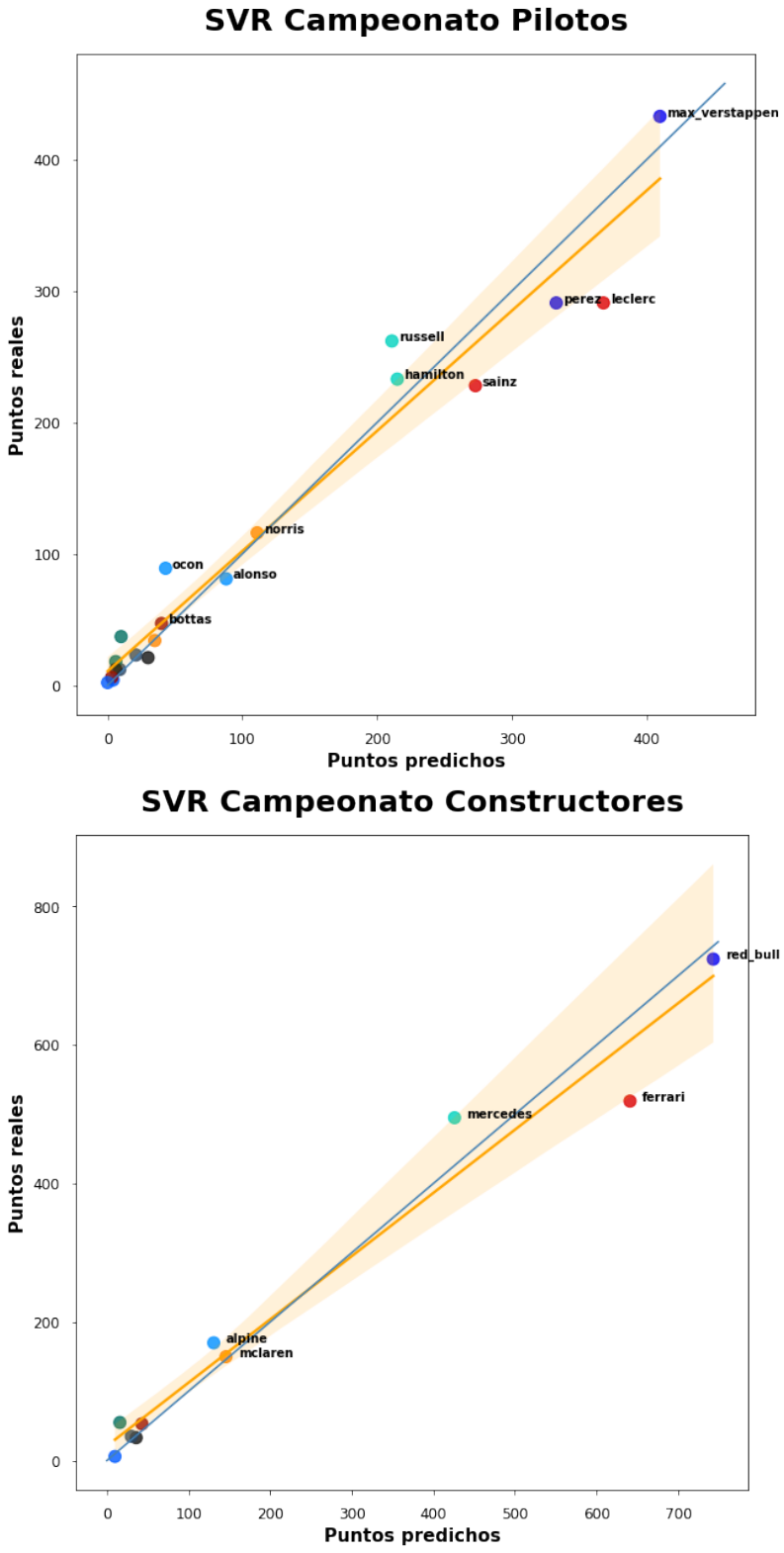


Figura D.2: SVR: gráficas de puntos

Apéndice E

Resultados RNA

	Piloto	Puntos predichos	Puntos reales	Posición predicha	Posición real	Error: (Pred - Real)
0	max_verstappen	438	433	1	1	0
1	leclerc	378	291	2	2	0
2	perez	308	291	3	2	1
3	sainz	289	228	4	6	-2
4	russell	244	262	5	4	1
5	hamilton	201	233	6	5	1
6	norris	110	116	7	7	0
7	alonso	76	81	8	9	-1
8	ocon	49	89	9	8	1
9	bottas	39	47	10	10	0
10	ricciardo	31	34	11	12	-1
11	kevin_magnussen	20	21	12	14	-2
12	gasly	15	23	13	13	0
13	vettel	8	37	14	11	3
14	stroll	4	18	15	15	0
15	tsunoda	4	12	15	16	-1
16	mick_schumacher	3	12	17	16	1
17	albon	2	4	18	19	-1
18	zhou	2	6	18	18	0
19	latifi	0	2	20	20	0

(a) Campeonato de Pilotos

	Constructor	Puntos predichos	Puntos reales	Posición predicha	Posición real	Error: (Pred - Real)
0	red_bull	746	724	1	1	0
1	ferrari	667	519	2	2	0
2	mercedes	445	495	3	3	0
3	mclaren	141	150	4	5	-1
4	alpine	125	170	5	4	1
5	alfa_romeo	41	53	6	7	-1
6	haas	23	33	7	9	-2
7	alpha_tauri	19	35	8	8	0
8	aston_martin	12	55	9	6	3
9	williams	3	6	10	10	0

(b) Campeonato de Constructores

Figura E.1: RNA: tablas de predicciones

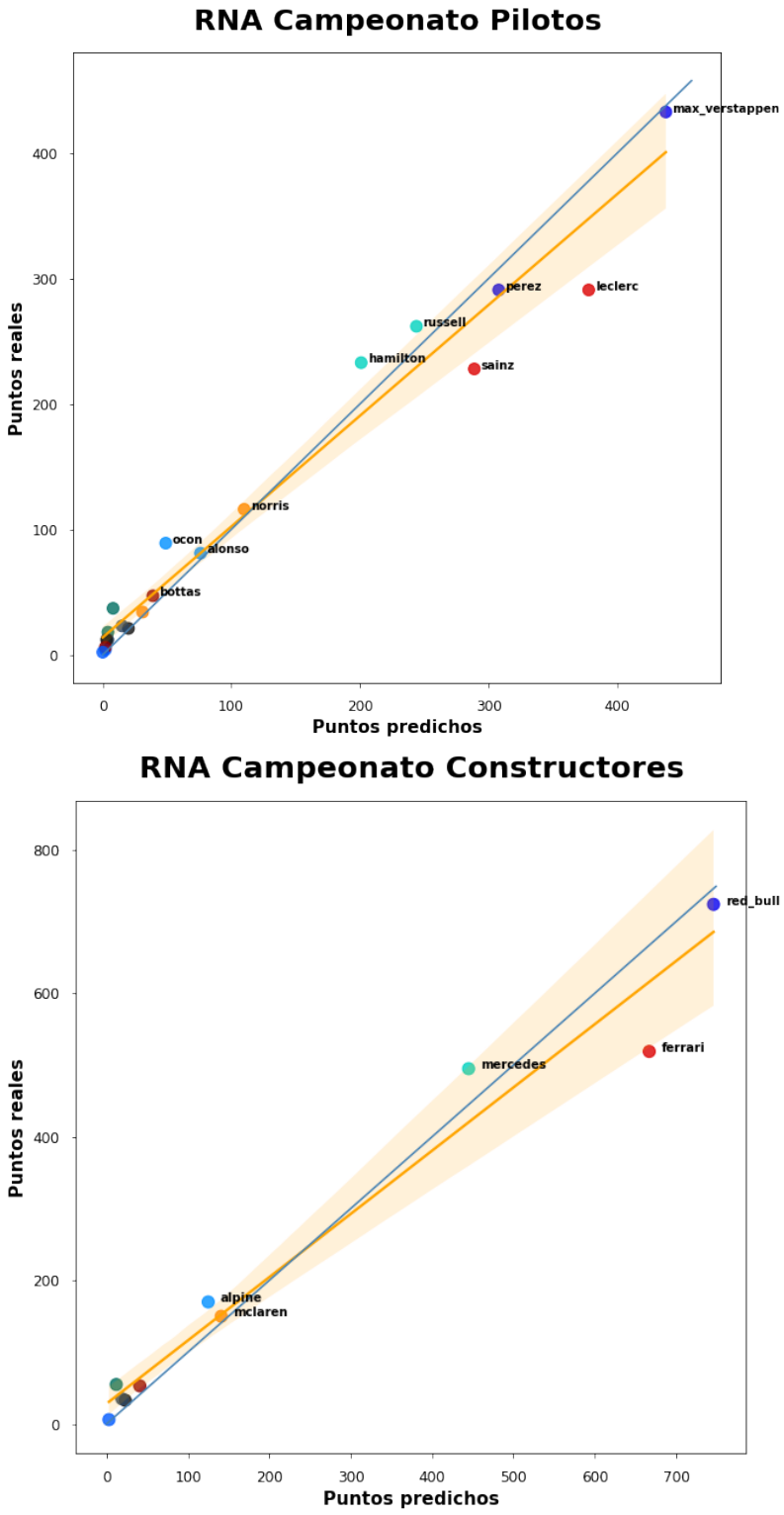


Figura E.2: RNA: gráficas de puntos

Apéndice F

Aplicación a la temporada 2023

Una vez observados los resultados logrados sobre la temporada 2022, surge la duda sobre la capacidad de predicción de los modelos sobre la siguiente temporada. El método a seguir (explicado en la sección 1.2) implica conocer la información previa a la carrera en cuestión, por lo que no es posible predecir todas las carreras hasta el fin de temporada. No obstante, se puede ir prediciendo cada carrera una vez conocida la información de la clasificación.

Para comprobar que los modelos de SL utilizados obtienen unos resultados de predicción parecidos independientemente de la temporada de estudio, se pueden analizar las primeras carreras de la temporada 2023. Para ello se tendrán en cuenta únicamente las carreras disputadas antes de la finalización del plazo de entrega del TFG (mediados de junio de 2023), dando un total de 7 carreras: Baréin, Arabia Saudí, Australia, Azerbaiyán, Miami, Mónaco y España.

Siguiendo la misma metodogía que la empleada en la sección 7.2, se procede a agregar los puntos de las predicciones de las distintas carreras para posteriormente comparar los puntos y posiciones con lo ocurrido en la realidad. Para el Campeonato de Pilotos de 2023, los resultados se pueden observar en la tabla F.1.

Modelo	*Spearman*	R^2	*MSE*	*RMSE*
RNA	0.929	0.917	164.80	12.837
SVR	0.911	0.905	189.40	13.762
XGB	0.924	0.875	249.15	15.784
RFR	0.911	0.859	280.50	16.748
DTR	0.886	0.743	510.80	22.601

Tabla F.1: Resultados de los modelos sobre el Campeonato de Pilotos de 2023

El mejor modelo vuelve a ser la RNA, seguido de cerca por el SVR. De igual manera que sucedía en la temporada 2022, el peor modelo a la hora de la predicciones resulta ser el DTR.

En la figura F.1 se pueden observar los puntos y posiciones, comparando las predicciones del mejor modelo (RNA) con la realidad.

	Piloto	Puntos predichos	Puntos reales	Posición predicha	Posición real	Error: (Pred - Real)
0	max_verstappen	132	164	1	1	0
1	perez	108	109	2	2	0
2	alonso	94	96	3	3	0
3	sainz	94	54	3	6	-3
4	hamilton	65	85	5	4	1
5	russell	60	60	6	5	1
6	leclerc	46	35	7	7	0
7	stroll	32	34	8	8	0
8	ocon	26	25	9	9	0
9	norris	18	12	10	11	-1
10	gasly	16	15	11	10	1
11	kevin_magnussen	8	2	12	16	-4
12	tsunoda	4	2	13	16	-3
13	hulkenberg	2	6	14	12	2
14	albon	1	1	15	18	-3
15	piastri	1	5	15	13	2
16	sargeant	0	0	17	19	-2
17	de_vries	0	0	17	19	-2
18	bottas	0	4	17	14	3
19	zhou	0	4	17	14	3

Figura F.1: Mejor predicción obtenida para Pilotos 2023

El modelo es capaz de predecir con bastante exactitud los puntos obtenidos por cada piloto en este primer tramo de la temporada, logrando pronosticar correctamente los tres primeros pilotos del campeonato. También se puede observar gráficamente la disposición de los puntos predichos y los puntos reales obtenidos en la figura F.2.

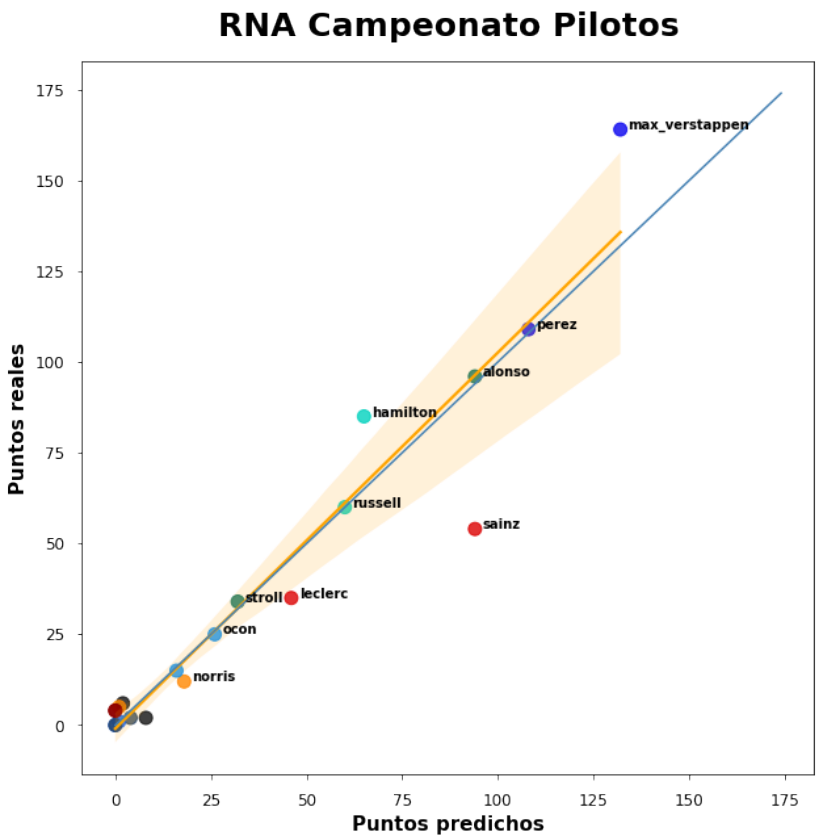


Figura F.2: Gráfica de puntos del mejor modelo para Pilotos 2023

Se observa que los pilotos se encuentran relativamente cerca de la línea ideal de predicción (línea azul). Gracias a la no inclusión de las variables de *"piloto"* y *"constructor"*, los modelos pueden predecir correctamente casos como el que sucede con Aston Martin, pasando de ser el 7º equipo la temporada pasada, a consolidarse como 3º mejor equipo hasta la fecha en 2023 gracias a los consecutivos podios de Alonso. Otras observaciones que se pueden obtener con la figura es el gran dominio de Verstappen en el campeonato con respecto a su compañero de equipo, además de los pocos puntos logrados por Ferrari (al igual que sucedía en la temporada 2022).

A continuación se muestra la tabla de predicciones (F.3) y la gráfica de puntos (F.4) correspondientes al peor modelo de predicción (DTR) para el Campeonato de Pilotos de 2023.

	Piloto	Puntos predichos	Puntos reales	Posición predicha	Posición real	Error: (Pred - Real)
0	alonso	108	96	1	3	-2
1	perez	106	109	2	2	0
2	hamilton	91	85	3	4	-1
3	sainz	88	54	4	6	-2
4	max_verstappen	84	164	5	1	4
5	russell	67	60	6	5	1
6	leclerc	66	35	7	7	0
7	ocon	50	25	8	9	-1
8	stroll	34	34	9	8	1
9	norris	33	12	10	11	-1
10	albon	13	1	11	18	-7
11	kevin_magnussen	12	2	12	16	-4
12	hulkenberg	12	6	12	12	0
13	piastri	8	5	14	13	1
14	gasly	8	15	14	10	4
15	tsunoda	7	2	16	16	0
16	de_vries	0	0	17	19	-2
17	sargeant	0	0	17	19	-2
18	bottas	0	4	17	14	3
19	zhou	0	4	17	14	3

Figura F.3: Peor predicción obtenida para Pilotos 2023

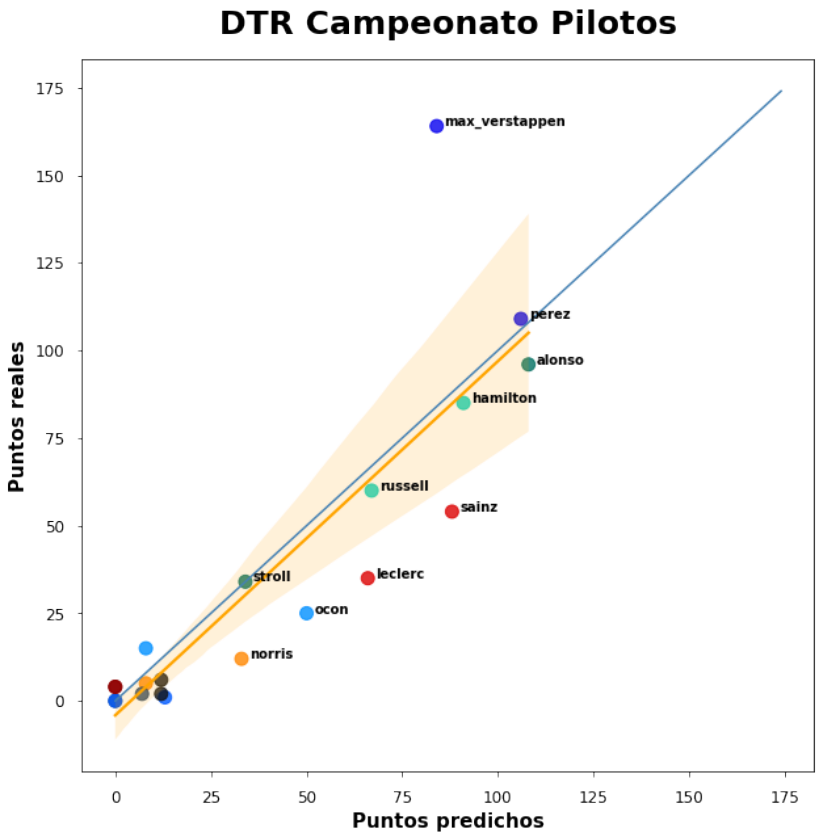


Figura F.4: Gráfica de puntos del peor modelo para Pilotos 2023

Este modelo no es capaz de predecir correctamente los tres primeros pilotos del campeonato y además, observando los puntos de la gráfica, se contempla que se encuentran fuera del intervalo de confianza de la regresión (área naranja). Aun así, el dominio de Verstappen y el bajo rendimiento del equipo Ferrari siguen presentes en esta predicción.

Para el caso del Campeonato de Constructores se sigue el mismo guión pero aplicado a los equipos. Las métricas obtenidas en el inicio de la temporada 2023 se pueden observar en la tabla F.2.

Modelo	***Spearman***	R^2	***MSE***	***RMSE***
RNA	0.900	0.942	418.6	20.460
SVR	0.900	0.908	658.8	25.667
XGB	0.930	0.878	871.7	29.525
RFR	0.881	0.860	999.2	31.610
DTR	0.924	0.821	1281.6	35.799

Tabla F.2: Resultados de los modelos sobre el Campeonato de Constructores de 2023

De nuevo, el mejor modelo es la RNA y el peor es el DTR. En la figura F.5 se pueden observar las mejores predicciones de posiciones y puntos de los diferentes constructores para la primera fase de la temporada 2023.

	Constructor	Puntos predichos	Puntos reales	Posición predicha	Posición real	Error: (Pred - Real)
0	red_bull	240	273	1	1	0
1	ferrari	140	89	2	4	-2
2	aston_martin	126	130	3	3	0
3	mercedes	125	145	4	2	2
4	alpine	42	40	5	5	0
5	mclaren	19	17	6	6	0
6	haas	10	8	7	7	0
7	alpha_tauri	4	2	8	9	-1
8	williams	1	1	9	10	-1
9	alfa_romeo	0	8	10	7	3

Figura F.5: Mejor predicción obtenida para Constructores 2023

El equipo líder sigue siendo Red Bull, seguido de Ferrari según el modelo, obteniendo así un error de dos posiciones en la predicción de la posición en el campeonato, debido a los numerosos accidentes, fallos de estrategia y bajo rendimiento en carrera que no les permitieron obtener un mayor botín de puntos. La predicción de puntos para el resto de equipos es bastante acertada. Gráficamente se puede ver en la figura F.6.

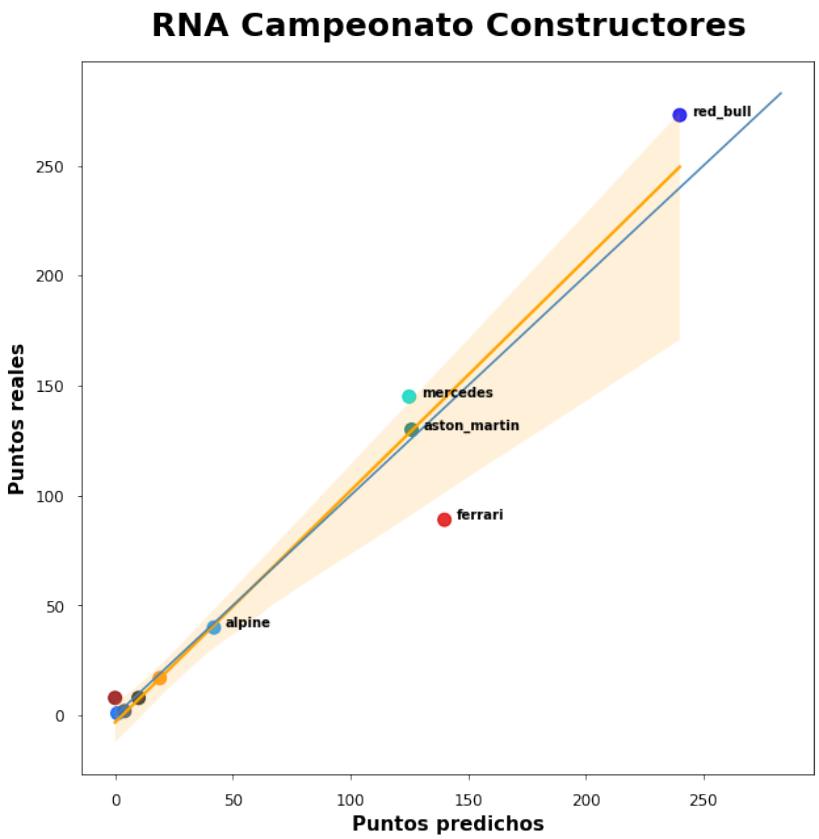


Figura F.6: Gráfica de puntos del mejor modelo para Constructores 2023

De igual manera que se lleva haciendo a lo largo de esta memoria, se compara el resultado del mejor modelo con el de peor rendimiento, que en este caso vuelve a ser el DTR. En las figuras F.7 y F.8 se pueden ver las predicciones obtenidas por el modelo.

	Constructor	Puntos predichos	Puntos reales	Posición predicha	Posición real	Error: (Pred - Real)
0	red_bull	190	273	1	1	0
1	mercedes	158	145	2	2	0
2	ferrari	154	89	3	4	-1
3	aston_martin	142	130	4	3	1
4	alpine	58	40	5	5	0
5	mclaren	41	17	6	6	0
6	haas	24	8	7	7	0
7	williams	13	1	8	10	-2
8	alpha_tauri	7	2	9	9	0
9	alfa_romeo	0	8	10	7	3

Figura F.7: Peor predicción obtenida para Constructores 2023

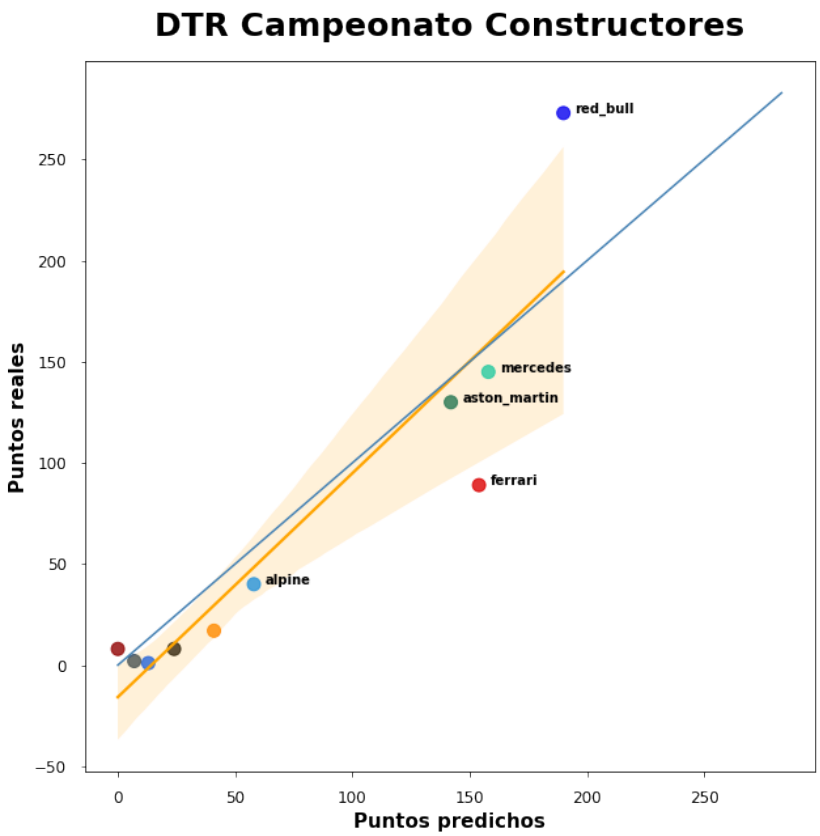


Figura F.8: Gráfica de puntos del peor modelo para Constructores 2023

Destacan los puntos asociados a Red Bull y a Ferrari, ya que los puntos predichos distan bastante de los puntos obtenidos en la realidad.

A modo de conclusión, de igual manera que sucede con las predicciones de la temporada 2022, en estas pocas carreras de 2023 los resultados obtenidos por los modelos son satisfactorios. Tanto para el caso de Pilotos como de Constructores, el mejor modelo ha sido la RNA y el peor el DTR.

Lista de acrónimos

AB Adaptive Boosting. 7, 8

CRISP-DM Cross Industry Standard Process for Data Mining. 14, 15

DT Decision Tree. 5, 6

DTR Decision Tree Regressor. 33, 40, 43, 45, 50, 52, 66, 67, 69–71

F1 Fórmula 1. 1, 4, 12, 13, 16, 21, 23, 25, 28, 51–53

FIA Federación Internacional de Automovilismo. 4, 29

GB Gradient Boosting. 7, 8

GP Grand Prix. 4, 13

ML Machine Learning. 5, 15, 16, 25, 34, 47, 51

MSE Mean Squared Error. 33

RF Random Forest. 6, 7, 13

RFR Random Forest Regressor. 33, 45, 50, 52

RMSE Root Mean Squared Error. 33, 38, 42, 44, 45, 51

RNA Red de Neuronas Artificiales. 9–11, 13, 33, 36, 38, 40, 42, 45–47, 50–53, 66, 69, 71

SL Supervised Learning. 2, 5, 32, 51, 65

SRP-CRISP-DM Sports Results Prediction CRISP-DM. 15

SVM Support Vector Machine. 8, 9, 13

SVR Support Vector Regressor. 9, 33, 42, 45–47, 51, 66

XGB Extreme Gradient Boosting. 8, 13, 33, 45, 48–50, 52

Bibliografía

[1] M. Mariani, "Guía de la Fórmula 1 para principiantes: sistema de puntuación, cómo funciona el Sprint, salarios, reglas de paradas en boxes y más," 2023. [En línea]. Disponible en: https://www.sportingnews.com/es/formula-1/news/guia-de-la-formula-1-para-principiantes-sistema-puntuacion-sprint-salarios-paradas-boxes-mas/jbf1nmywxgn4rxbwcn4htakp

[2] A. S. Martin, "Fórmula 1 para principiantes: lo que debes saber sobre la máxima categoría del automovilismo," 2023. [En línea]. Disponible en: https://businessinsider.mx/f1-para-principiantes-lo-que-debes-saber-para-entenderla-guia-basica/

[3] J.R.Quinlan, *Induction of decision trees*, 1st ed. Machine Learning, 1986.

[4] "¿Qué es un árbol de decisión?" [En línea]. Disponible en: https://www.ibm.com/es-es/topics/decision-trees

[5] "Random Forest: Bosque aleatorio. Definición y funcionamiento," 2022. [En línea]. Disponible en: https://datascientest.com/es/random-forest-bosque-aleatorio-definicion-y-funcionamiento

[6] J. M. Heras, "Ensembles: voting, bagging, boosting, stacking," 2019. [En línea]. Disponible en: https://www.iartificial.net/ensembles-voting-bagging-boosting-stacking/

[7] "¿Qué es bagging?" [En línea]. Disponible en: https://www.ibm.com/es-es/topics/bagging

[8] "¿Qué es boosting?" [En línea]. Disponible en: https://www.ibm.com/es-es/topics/boosting

[9] A. Choudhury, "What is Gradient Boosting? How is it different from Ada Boost?" 2020. [En línea]. Disponible en: https://medium.com/analytics-vidhya/what-is-gradient-boosting-how-is-it-different-from-ada-boost-2d5ff5767cb2

[10] "XGBoost Documentation." [En línea]. Disponible en: https://xgboost.readthedocs.io/en/stable/

[11] M. Mariani, "A Brief Introduction to XGBoost," 2020. [En línea]. Disponible en: https://towardsdatascience.com/a-brief-introduction-to-xgboost-3eaee2e3e5d6

[12] J. A. Rodrigo, "Máquinas de Vector Soporte (Support Vector Machines, SVMs)," 2017. [En línea]. Disponible en: https://www.cienciadedatos.net/documentos/34_maquinas_de_vector_soporte_support_vector_machines

[13] T. Sharp, "An Introduction to Support Vector Regression (SVR)," 2020. [En línea]. Disponible en: https://towardsdatascience.com/an-introduction-to-support-vector-regression-svr-a3ebc1672c2

[14] "Qué son las redes neuronales y sus funciones," 2022. [En línea]. Disponible en: Quésonlasredesneuronalesysusfunciones

[15] J. Torres, "Learning Process of a Deep Neural Network," 2020. [En línea]. Disponible en: https://towardsdatascience.com/learning-process-of-a-deep-neural-network-5a9768d7a651

[16] B. Ofoghi, J. Zeleznikow, C. Macmahon, J. Rehula, and D. Dwyer, "Performance analysis and prediction in triathlon," *Journal of sports sciences*, vol. 34, pp. 1–6, 07 2015. [En línea]. Disponible en: https://doi.org/10.1080/02640414.2015.1065341

[17] R. Eichenberger and D. Stadelmann, "Who Is The Best Formula 1 Driver? An Economic Approach to Evaluating Talent," *Economic Analysis and Policy*, vol. 1, 12 2009. [En línea]. Disponible en: https://doi.org/10.1016/S0313-5926(09)50035-5

[18] A. Bell, J. Smith, C. Sabel, and K. Jones, "Formula for success: Multilevel modelling of Formula One Driver and Constructor performance, 1950-2014," *Journal of Quantitative Analysis in Sports*, vol. 12, pp. 99–112, 06 2016. [En línea]. Disponible en: https://doi.org/10.1515/jqas-2015-0050

[19] E.-J. van Kesteren and T. Bergkamp, "Bayesian Analysis of Formula One Race Results: Disentangling Driver Skill and Constructor Advantage," 2022. [En línea]. Disponible en: https://doi.org/10.48550/arxiv.2203.08489

[20] R. Bol, "How to win in Formula One: is it the driver or the car?" 2020. [En línea]. Disponible en: https://thecorrespondent.com/642/how-to-win-in-formula-one-is-it-the-driver-or-the-car

[21] E. Stoppels, "Predicting race results using artificial neural networks," 2017.

[22] V. Nigro, "Formula 1 Race Predictor," 2020. [En línea]. Disponible en: https://towardsdatascience.com/formula-1-race-predictor-5d4bfae887da

[23] W. George, "Formula 1 Championship Predictor," 2021. [En línea]. Disponible en: https://medium.com/@willgeorge93/formula-1-championship-predictor-a-machine-learning-solution-a86efcb9298

[24] J. F. V. Rueda, "CRISP-DM: una metodología para minería de datos en salud," 2019. [En línea]. Disponible en: https://healthdataminer.com/data-mining/crisp-dm-una-metodologia-para-mineria-de-datos-en-salud/

[25] R. Bunker and F. Thabtah, "A Machine Learning Framework for Sport Result Prediction," *Applied Computing and Informatics*, vol. 15, 2017. [En línea]. Disponible en: https://doi.org/10.1016/j.aci.2017.09.005

[26] "Python." [En línea]. Disponible en: https://www.python.org

[27] "Visual Studio Code." [En línea]. Disponible en: https://code.visualstudio.com

[28] "Pandas." [En línea]. Disponible en: https://pandas.pydata.org

[29] "NumPy." [En línea]. Disponible en: https://numpy.org

[30] "Requests." [En línea]. Disponible en: https://pypi.org/project/requests/

[31] "Seaborn." [En línea]. Disponible en: https://seaborn.pydata.org

[32] "Matplotlib." [En línea]. Disponible en: https://matplotlib.org

[33] "Dython." [En línea]. Disponible en: https://pypi.org/project/dython/

[34] "Scikit-learn." [En línea]. Disponible en: https://scikit-learn.org/stable/

[35] "xgboost." [En línea]. Disponible en: https://xgboost.readthedocs.io/en/stable/

[36] "Ergast API." [En línea]. Disponible en: http://ergast.com/mrd/

[37] "Coulthard ready to start again in Sepang." 2005. [En línea]. Disponible en: https://www.crash.net/f1/news/51485/1/coulthard-ready-to-start-again-in-sepang

[38] J. Terenzio, "Finding the Limit - Formula 1 Data Visualizations and Points Prediction," 2021. [En línea]. Disponible en: https://julianterenzio.io/blog/Finding%20the%20Limit%20-%20Formula%201%20Data%20Visualizations%20and%20Points%20Prediction

[39] "¿Qué es una Matriz de Correlación?" 2020. [En línea]. Disponible en: https://datascience.eu/es/matematica-y-estadistica/que-es-una-matriz-de-correlacion/

[40] S. Zychlinski, "The Search for Categorical Correlation - Towards Data Science," 2018. [En línea]. Disponible en: https://towardsdatascience.com/the-search-for-categorical-correlation-a1cf7f1888c9

[41] "An overview of correlation measures between categorical and continuous variables," 2018. [En línea]. Disponible en: https://medium.com/@outside2SDs/an-overview-of-correlation-measures-between-categorical-and-continuous-variables-4c7f85610365

[42] B. Jones, *Formula One Circuits from Above 2022.* Welbeck Publishing, 2022.

[43] G. Echeverria, "Nuevo reglamento de Fórmula 1 2014," 2014. [En línea]. Disponible en: https://www.motor.es/formula-1/nuevo-reglamento-de-formula-1-2014.php

[44] D. Plaza, "Reglamento F1 2022: esto es todo lo que cambia en los coches," 2022. [En línea]. Disponible en: https://www.motor.es/formula-1/reglamento-f1-2022-novedades-202283954.html

[45] N. A. L. Cosio, "Métricas en regresión - Nicolás Arrioja Landa Cosio," 2022. [En línea]. Disponible en: https://medium.com/@nicolasarrioja/métricas-en-regresión-5e5d4259430b

[46] A. Gupta, "Spearman's Rank Correlation: The Definitive Guide To Understand," 2023. [En línea]. Disponible en: https://www.simplilearn.com/tutorials/statistics-tutorial/spearmans-rank-correlation

[47] P. Nellihela, "What is K-fold Cross Validation? - Towards Data Science," 2022. [En línea]. Disponible en: https://towardsdatascience.com/what-is-k-fold-cross-validation-5a7bb241d82f

[48] "Importancia de la característica de permutación: referencia de componente - Azure Machine Learning," 2022. [En línea]. Disponible en: https://learn.microsoft.com/es-es/azure/machine-learning/component-reference/permutation-feature-importance

[49] T. Jensen, "Feature Importance for Any Model using Permutation - Taylor Jensen," 2022. [En línea]. Disponible en: https://medium.com/@T_Jen/feature-importance-for-any-model-using-permutation-7997b7287aa

[50] "F1 Insights con tecnología de AWS | Fórmula 1 usa Amazon Web Services." [En línea]. Disponible en: https://aws.amazon.com/es/sports/f1/